QuickTime Streaming Reference

QuickTime > Streaming

2006-05-23

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

QuickTime Streaming Reference

Framework:	Frameworks/QuickTime.framework
Declared in	ImageCompression.h Movies.h QTSMovie.h QTStreamingComponents.h QuickTimeStreaming.h

Overview

The streaming API in QuickTime allows developers to recognize and play streaming movies, add hint tracks so movies can be streamed, create packetizers and reassemblers, mix streaming and nonstreaming data in a single movie, and broadcast live streams in real time.

Functions

DisposeQTSModalFilterUPP

Disposes of a QTSModalFilterUPP pointer.

```
void DisposeQTSModalFilterUPP (
   QTSModalFilterUPP userUPP
);
```

Parameters

userUPP

A `QTSModalFilterUPP` pointer.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

DisposeQTSNotificationUPP

Disposes of a QTSNotificationUPP pointer.

```
void DisposeQTSNotificationUPP (
   QTSNotificationUPP userUPP
);
```

Parameters

userUPP

A `QTSNotificationUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

DisposeQTSPanelFilterUPP

Disposes of a QTSPanelFilterUPP pointer.

```
void DisposeQTSPanelFilterUPP (
   QTSPanelFilterUPP userUPP
);
```

Parameters

userUPP

A `QTSPanelFilterUPP` pointer.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.1 and later.

Declared In
`QuickTimeStreaming.h`

DisposeRTPMPDataReleaseUPP

Disposes of an RTPMPDataReleaseUPP pointer.

```
void DisposeRTPMPDataReleaseUPP (
   RTPMPDataReleaseUPP userUPP
);
```

Parameters

userUPP

An `RTPMPDataReleaseUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

DisposeRTPPBCallbackUPP

Disposes of an RTPPBCallbackUPP pointer.

```
void DisposeRTPPBCallbackUPP (
   RTPPBCallbackUPP userUPP
);
```

Parameters

`userUPP`

An `RTPPBCallbackUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

InitializeQTS

Initializes QuickTime streaming.

```
OSErr InitializeQTS (
   void
);
```

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

NewQTSModalFilterUPP

Allocates a Universal Procedure Pointer for the QTSModalFilterProc callback.

```
QTSModalFilterUPP NewQTSModalFilterUPP (
   QTSModalFilterProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

NewQTSNotificationUPP

Allocates a Universal Procedure Pointer for the QTSNotificationProc callback.

```
QTSNotificationUPP NewQTSNotificationUPP (
   QTSNotificationProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

Discussion

This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes

Introduced in QuickTime 4.1. Replaces `NewQTSNotificationProc`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

NewQTSPanelFilterUPP

Allocates a Universal Procedure Pointer for the QTSPanelFilterProc callback.

```
QTSPanelFilterUPP NewQTSPanelFilterUPP (
   QTSPanelFilterProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.1 and later.

Declared In
`QuickTimeStreaming.h`

NewRTPMPDataReleaseUPP

Allocates a Universal Procedure Pointer for the RTPMPDataReleaseProc callback.

```
RTPMPDataReleaseUPP NewRTPMPDataReleaseUPP (
   RTPMPDataReleaseProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewRTPMPDataReleaseProc`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

NewRTPPBCallbackUPP

Allocates a Universal Procedure Pointer for the RTPPBCallbackProc callback.

```
RTPPBCallbackUPP NewRTPPBCallbackUPP (
   RTPPBCallbackProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

Discussion

This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes

Introduced in QuickTime 4.1. Replaces `NewRTPPBCallbackProc`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

QTSAllocBuffer

Allocates a QuickTime streaming stream buffer.

```
QTSStreamBuffer * QTSAllocBuffer (
   SInt32 inSize
);
```

Parameters

inSize

The size of the buffer to be allocated.

Return Value

A `QTSStreamBuffer` structure

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSAllocMemPtr

Undocumented

```
QTSMemPtr QTSAllocMemPtr (
   UInt32 inByteCount,
   SInt32 inFlags
);
```

Parameters

inByteCount

Undocumented

inFlags

Undocumented

Return Value
Undocumented

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSCopyMessage

Undocumented

```
QTSStreamBuffer * QTSCopyMessage (
   QTSStreamBuffer *inMessage
);
```

Parameters

inMessage

A pointer to a `QTSStreamBuffer` structure.

Return Value
A pointer to a `QTSStreamBuffer` structure.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSDisposePresentation

Disposes of a QuickTime streaming presentation.

```
OSErr QTSDisposePresentation (
   QTSPresentation inPresentation,
   SInt32 inFlags
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines the presentation to be disposed.

inFlags

Flags governing the disposal of the presentation. Currently, no flags are defined; set this parameter to 0.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSDisposeStatHelper

Disposes of a QuickTime streaming statistics helper that was previously created by QTSNewStatHelper.

```
OSErr QTSDisposeStatHelper (
   QTSStatHelper inStatHelper
);
```

Parameters

inStatHelper

A pointer to a `QTSStatHelperRecord` structure that defines the statistics helper to be disposed.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSDisposeStream

Disposes of a QuickTime streaming stream.

```
OSErr QTSDisposeStream (
   QTSStream inStream,
   SInt32 inFlags
);
```

Parameters

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream to be disposed.

inFlags

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSDuplicateMessage

Undocumented

```
OSErr QTSDuplicateMessage (
   QTSStreamBuffer *inMessage,
   SInt32 inFlags,
   QTSStreamBuffer **outDuplicatedMessage
);
```

Parameters

inMessage

Undocumented

inFlags

Undocumented

outDuplicatedMessage

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSDupMessage

Undocumented

```
QTSStreamBuffer * QTSDupMessage (
   QTSStreamBuffer *inMessage
);
```

Parameters

inMessage

A pointer to a `QTSStreamBuffer` structure.

Return Value
A pointer to a `QTSStreamBuffer` structure.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSFindMediaPacketizer

Creates a list of media packetizers that can work with a specified sample description and meet specified criteria.

```
OSErr QTSFindMediaPacketizer (
   MediaPacketizerRequirementsPtr inPacketizerinfo,
   SampleDescriptionHandle inSampleDescription,
   RTPPayloadSortRequestPtr inSortInfo,
   QTAtomContainer *outPacketizerList
);
```

Parameters

inPacketizerinfo

A pointer to a `MediaPacketizerRequirements` structure that specifies the required features of the media packetizers you are looking for.

inSampleDescription

A handle to a `SampleDescription` structure that specifies the media data the packetizer needs to work with.

inSortInfo

A pointer to a `RTPPayloadSortRequest` structure that specifies the sort order for the list of packetizers.

outPacketizerList

On entry, a pointer to a handle to a QT atom container. On return, this container will be filled with a sorted list of available media packetizers that meet the specified criteria. Only packetizers that have the features specified by `inPacketizerInfo` will be listed. The list will be sorted in the order specified by `inSortInfo`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSFindMediaPacketizerForPayloadID

Creates a list of media packetizers for a specified payload number.

```
OSErr QTSFindMediaPacketizerForPayloadID (
   long payloadID,
   RTPPayloadSortRequestPtr inSortInfo,
   QTAtomContainer *outPacketizerList
);
```

Parameters

`payloadID`

An IETF payload number.

`inSortInfo`

A pointer to a `RTPPayloadSortRequest` structure that specifies the sort order for the list of packetizers.

`outPacketizerList`

On entry, a pointer to a handle to a QT atom container. On return, this container will be filled with a sorted list of available media packetizers for the specified payload ID. The list will be sorted in the order specified by `inSortInfo`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSFindMediaPacketizerForPayloadName

Creates a list of media packetizers for a specified payload name.

```
OSErr QTSFindMediaPacketizerForPayloadName (
   const char *payloadName,
   RTPPayloadSortRequestPtr inSortInfo,
   QTAtomContainer *outPacketizerList
);
```

Parameters

payloadName

A pointer to a payload name string.

inSortInfo

A pointer to a `RTPPayloadSortRequest` structure that specifies the sort order for the list of packetizers.

outPacketizerList

On entry, a pointer to a handle to a QT atom container. On return, this container will be filled with a sorted list of available media packetizers for the specified payload name. The list will be sorted in the order specified by `inSortInfo`.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

QTSFindMediaPacketizerForTrack

Creates a list of media packetizers for a specified movie track and sample data.

```
OSErr QTSFindMediaPacketizerForTrack (
   Track inTrack,
   long inSampleDescriptionIndex,
   RTPPayloadSortRequestPtr inSortInfo,
   QTAtomContainer *outPacketizerList
);
```

Parameters

inTrack

The track for this operation. Your application obtains this track identifier from such functions as `NewMovieTrack` and `GetMovieTrack`.

inSampleDescriptionIndex

The value of the `dataRefIndex` field of the `SampleDescription` structure that specifies the type of media data that will be packetized.

inSortInfo

A pointer to a `RTPPayloadSortRequest` structure that specifies the sort order for the list of packetizers.

outPacketizerList

On entry, a pointer to a handle to a QT atom container. On return, this container will be filled with a sorted list of available media packetizers for the specified track. The list will be sorted in the order specified by `inSortInfo`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSFindReassemblerForPayloadID

Creates a list of streaming reassemblers for a specified payload number.

```
OSErr QTSFindReassemblerForPayloadID (
   UInt8 inPayloadID,
   RTPPayloadSortRequest *inSortInfo,
   QTAtomContainer *outReassemblerList
);
```

Parameters

inPayloadID

An IETF payload number.

inSortInfo

A pointer to a `RTPPayloadSortRequest` structure that specifies the sort order for the list of reassemblers.

outReassemblerList

On entry, a pointer to a handle to a QT atom container. On return, this container will be filled with a sorted list of available reassemblers for the specified track. The list will be sorted in the order specified by `inSortInfo`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSFindReassemblerForPayloadName

Creates a list of streaming reassemblers for a specified payload name.

```
OSErr QTSFindReassemblerForPayloadName (
   const char *inPayloadName,
   RTPPayloadSortRequest *inSortInfo,
   QTAtomContainer *outReassemblerList
);
```

Parameters

inPayloadName

A payload name string.

inSortInfo

A pointer to a `RTPPayloadSortRequest` structure that specifies the sort order for the list of reassemblers.

outReassemblerList

On entry, a pointer to a handle to a QT atom container. On return, this container will be filled with a sorted list of available reassemblers for the specified track. The list will be sorted in the order specified by `inSortInfo`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSFlattenMessage

Undocumented

```
QTSStreamBuffer * QTSFlattenMessage (
   QTSStreamBuffer *inMessage
);
```

Parameters

inMessage

A pointer to a `QTSStreamBuffer` structure.

Return Value
A pointer to a `QTSStreamBuffer` structure.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSFreeMessage

Undocumented

```
void QTSFreeMessage (
   QTSStreamBuffer *inMessage
);
```

Parameters

inMessage

A pointer to a `QTSStreamBuffer` structure.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSGetErrorString

Undocumented

```
Boolean QTSGetErrorString (
   SInt32 inErrorCode,
   UInt32 inMaxErrorStringLength,
   char *outErrorString,
   SInt32 inFlags
);
```

Parameters

inErrorCode

Undocumented

inMaxErrorStringLength

Undocumented

outErrorString

Undocumented

inFlags

Undocumented

Return Value
Undocumented

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSGetNetworkAppName

Gets the name of a streaming network application.

```
OSErr QTSGetNetworkAppName (
   SInt32 inFlags,
   char **outCStringPtr
);
```

Parameters

inFlags

A flag (see below) that determines whether the application name is a full pathname. See these constants:

`kQTSNetworkAppNameIsFullNameFlag`

outCStringPtr

A `Ptr` to a `CStringPtr`; see `MacTypes.h`. This information is sent back to servers in HTTP and RTSP headers, so they can work out client statistics. A typical default string is `QTS (qtver=4.1.1;cpu=PPC;os=Mac 9.0.4)`.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

Following is an example of calling this function:

```
Ptr networkAppName =NIL;
err =QTSGetNetworkAppName(0L, &networkAppName);
printf("The NetworkAppName is %s", networkAppName);
DisposePtr(networkAppName);
// This call prints
// The NetworkAppName is QTS (qtver=4.1.1;cpu=PPC;os=Mac 9.0.4)
// or
// The NetworkAppName is QTS (qtver=4.0;os=Windows NT 4.0 Service Pack 3)
// If you set it from your app, that will be returned instead.
```

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSGetOrMakeStatAtomForStream

Gets the statistics atom for a stream or creates a new statistics atom for it.

```
OSErr QTSGetOrMakeStatAtomForStream (
   QTAtomContainer inContainer,
   QTSStream inStream,
   QTAtom *outParentAtom
);
```

Parameters

inContainer

An atom container that holds the statistics atoms for the specified stream.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

outParentAtom

On entry, a pointer to a variable of type `QTAtom`; on return, this variable is set to the atom that holds the statistics for this stream. If no such atom exists for that stream, then the function creates a statistics atom.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is to be used only by stream components to put stream statistics into an atom container; applications should not call it.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSGetStreamPresentation

Gets the presentation for a stream.

```
QTSPresentation QTSGetStreamPresentation (
   QTSStream inStream
);
```

Parameters

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

Return Value
A pointer to a `QTSPresentationRecord` structure.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSInitializeMediaParams

Undocumented

```
OSErr QTSInitializeMediaParams (
   QTSMediaParams *inMediaParams
);
```

Parameters

inMediaParams

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.1 and later.

Declared In

`QuickTimeStreaming.h`

QTSInsertStatistic

Inserts statistics data into the statistic atom for a stream.

```
OSErr QTSInsertStatistic (
   QTAtomContainer inContainer,
   QTAtom inParentAtom,
   OSType inStatType,
   void *inStatData,
   UInt32 inStatDataLength,
   OSType inStatDataFormat,
   SInt32 inFlags
);
```

Parameters

inContainer

A handle to the atom container that contains the statistic atom.

inParentAtom

The atom that will hold a new atom containing the specified statistic data.

inStatType

A constant (see below) that identifies the type of statistic atom to insert the data into. See these constants:

`kQTSStatisticsStreamAtomType`
`kQTSStatisticsNameAtomType`
`kQTSStatisticsDataFormatAtomType`
`kQTSStatisticsDataAtomType`
`kQTSStatisticsUnitsAtomType`
`kQTSStatisticsUnitsNameAtomType`

inStatData

A pointer to a structure containing the data to insert.

inStatDataLength

The length, in bytes, of the statistic data.

inStatDataFormat

A constant (see below) that identifies the format of the inserted statistic atom. See these constants:

```
kQTSStatisticsSInt32DataFormat
kQTSStatisticsUInt32DataFormat
kQTSStatisticsSInt16DataFormat
kQTSStatisticsUInt16DataFormat
kQTSStatisticsFixedDataFormat
kQTSStatisticsStringDataFormat
kQTSStatisticsOSTypeDataFormat
```

inFlags

Currently no flags are defined; pass 0 in this parameter.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is to be used only by stream components to put stream statistics into an atom container; applications should not call it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSInsertStatisticName

Inserts the name and type of a statistic datum into the statistic atom for a stream.

```
OSErr QTSInsertStatisticName (
   QTAtomContainer inContainer,
   QTAtom inParentAtom,
   OSType inStatType,
   const char *inStatName,
   UInt32 inStatNameLength
);
```

Parameters

inContainer

A handle to the atom container that contains the statistic atom. Both the atom container and the parent atom must already exist.

inParentAtom

The atom that will hold a new atom containing the specified statistic name and type.

inStatType

A constant (see below) that identifies the type of statistic atom to insert the data into. See these constants:

```
kQTSStatisticsStreamAtomType
kQTSStatisticsNameAtomType
kQTSStatisticsDataFormatAtomType
kQTSStatisticsDataAtomType
kQTSStatisticsUnitsAtomType
kQTSStatisticsUnitsNameAtomType
```

inStatName

A pointer to the name string to be inserted.

inStatNameLength

The length of the name string in characters.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is to be used only by stream components to put stream statistics into an atom container; applications should not call it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSInsertStatisticUnits

Inserts the name and type of statistic units into the statistic atom for a stream.

```
OSErr QTSInsertStatisticUnits (
   QTAtomContainer inContainer,
   QTAtom inParentAtom,
   OSType inStatType,
   OSType inUnitsType,
   const char *inUnitsName,
   UInt32 inUnitsNameLength
);
```

Parameters

inContainer

A handle to the atom container that contains the statistic atom. Both the atom container and the parent atom must already exist.

inParentAtom

The atom that will hold a new atom containing the specified statistic name and type.

inStatType

A constant (see below) that identifies the type of statistic atom to insert the data into. See these constants:

```
kQTSStatisticsStreamAtomType
kQTSStatisticsNameAtomType
kQTSStatisticsDataFormatAtomType
kQTSStatisticsDataAtomType
kQTSStatisticsUnitsAtomType
kQTSStatisticsUnitsNameAtomType
```

inUnitsType

A constant (see below) that identifies the type of units atom to insert the data into. See these constants:

```
kQTSStatisticsNoUnitsType
kQTSStatisticsPercentUnitsType
kQTSStatisticsBitsPerSecUnitsType
kQTSStatisticsFramesPerSecUnitsType
```

inUnitsName

A pointer to the units name string to be inserted.

inUnitsNameLength

The length of the units name string in characters.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is to be used only by stream components to put stream statistics into an atom container; applications should not call it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSMediaGetIndStreamInfo

Undocumented

```
ComponentResult QTSMediaGetIndStreamInfo (
   MediaHandler mh,
   SInt32 inIndex,
   OSType inSelector,
   void *ioParams
);
```

Parameters

mh

A media handler. You can obtain this reference from `GetMediaHandler`.

inIndex

Undocumented

inSelector

A constant (see below) that identifies the type of information to be retrieved. See these constants:

```
kQTSMediaPresentationInfo
kQTSMediaNotificationInfo
kQTSMediaTotalDataRateInfo
kQTSMediaLostPercentInfo
kQTSMediaNumStreamsInfo
kQTSMediaIndSampleDescriptionInfo
```

ioParams

A pointer to returned information in a format determined by `inSelector` (see below).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTSMovie.h`

QTSMediaGetInfo

Gets information about a streaming media.

```
ComponentResult QTSMediaGetInfo (
   MediaHandler mh,
   OSType inSelector,
   void *ioParams
);
```

Parameters

mh

A media handler. You can obtain this reference from `GetMediaHandler`.

inSelector

A constant (see below) that identifies the type of information to be retrieved. See these constants:

```
kQTSMediaPresentationInfo
kQTSMediaNotificationInfo
kQTSMediaTotalDataRateInfo
kQTSMediaLostPercentInfo
kQTSMediaNumStreamsInfo
kQTSMediaIndSampleDescriptionInfo
```

ioParams

A pointer to returned information in a format determined by `inSelector` (see below).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTSMovie.h`

QTSMediaSetIndStreamInfo

Undocumented

```
ComponentResult QTSMediaSetIndStreamInfo (
   MediaHandler mh,
   SInt32 inIndex,
   OSType inSelector,
   void *ioParams
);
```

Parameters

mh

A media handler. You can obtain this reference from `GetMediaHandler`.

inIndex

Undocumented

inSelector

A constant (see below) that identifies the type of information to be set. See these constants:

`kQTSMediaPresentationInfo`
`kQTSMediaNotificationInfo`
`kQTSMediaTotalDataRateInfo`
`kQTSMediaLostPercentInfo`
`kQTSMediaNumStreamsInfo`
`kQTSMediaIndSampleDescriptionInfo`

ioParams

A pointer to information in a format determined by `inSelector` (see below).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTSMovie.h`

QTSMediaSetInfo

Sets information about a streaming media.

```
ComponentResult QTSMediaSetInfo (
   MediaHandler mh,
   OSType inSelector,
   void *ioParams
);
```

Parameters

mh

A media handler. You can obtain this reference from `GetMediaHandler`.

inSelector

A constant (see below) that identifies the type of information to be set. See these constants:

`kQTSMediaPresentationInfo`
`kQTSMediaNotificationInfo`
`kQTSMediaTotalDataRateInfo`
`kQTSMediaLostPercentInfo`
`kQTSMediaNumStreamsInfo`
`kQTSMediaIndSampleDescriptionInfo`

ioParams

A pointer to information in a format determined by `inSelector` (see below).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTSMovie.h`

QTSMessageLength

Undocumented

```
UInt32 QTSMessageLength (
   QTSStreamBuffer *inMessage
);
```

Parameters

inMessage

A pointer to a `QTSStreamBuffer` structure.

Return Value
The message length.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSNewHandle

Allocates a new handle for data, with options and checking.

```
Handle QTSNewHandle (
   UInt32 inByteCount,
   SInt32 inFlags,
   SInt32 *outFlags
);
```

Parameters

inByteCount

The requested size in bytes of the relocatable block.

inFlags

Flags (see below) that control memory allocation options. See these constants:

`kQTSMemAllocClearMem`
`kQTSMemAllocDontUseTempMem`
`kQTSMemAllocTryTempMemFirst`
`kQTSMemAllocDontUseSystemMem`
`kQTSMemAllocTrySystemMemFirst`
`kQTSMemAllocHoldMemory`
`kQTSMemAllocIsInterruptTime`

outFlags

A pointer to memory where return flags (see below) report on the block's actual memory location. See these constants:

`kQTSMemAllocAllocatedInTempMem`
`kQTSMemAllocAllocatedInSystemMem`

Return Value
The new handle.

Discussion
This function is a handy way to allocate memory without overflowing the application heap, which is mostly a concern with Mac OS versions 7 through 9. It is often used for streaming data.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSNewPresentation

Creates a new streaming presentation.

```
OSErr QTSNewPresentation (
   const QTSNewPresentationParams *inParams,
   QTSPresentation *outPresentation
);
```

Parameters

inParams

A pointer to a `QTSNewPresentationParams` structure that specifies the presentation.

outPresentation

A pointer to a pointer to a new `QTSPresentationRecord` structure.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSNewPresentationFromData

Undocumented

```
OSErr QTSNewPresentationFromData (
   OSType inDataType,
   const void *inData,
   const SInt64 *inDataLength,
   const QTSPresParams *inPresParams,
   QTSPresentation *outPresentation
);
```

Parameters

inDataType

Undocumented

inData

Undocumented

inDataLength

Undocumented

inPresParams

Undocumented

outPresentation

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSNewPresentationFromDataRef

Undocumented

```
OSErr QTSNewPresentationFromDataRef (
   Handle inDataRef,
   OSType inDataRefType,
   const QTSPresParams *inPresParams,
   QTSPresentation *outPresentation
);
```

Parameters

inDataRef
Undocumented

inDataRefType
Undocumented

inPresParams
Undocumented

outPresentation
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSNewPresentationFromFile

Undocumented

```
OSErr QTSNewPresentationFromFile (
   const FSSpec *inFileSpec,
   const QTSPresParams *inPresParams,
   QTSPresentation *outPresentation
);
```

Parameters

inFileSpec

Undocumented

inPresParams

Undocumented

outPresentation

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSNewPtr

Allocates a block of memory for streaming data, with options and checking, and returns a pointer to it.

```
Ptr QTSNewPtr (
   UInt32 inByteCount,
   SInt32 inFlags,
   SInt32 *outFlags
);
```

Parameters

inByteCount

The requested size in bytes of the new memory block.

inFlags

Flags (see below) that control memory allocation options. See these constants:

`kQTSMemAllocClearMem`

`kQTSMemAllocDontUseTempMem`

`kQTSMemAllocTryTempMemFirst`

`kQTSMemAllocDontUseSystemMem`

`kQTSMemAllocTrySystemMemFirst`

`kQTSMemAllocHoldMemory`

`kQTSMemAllocIsInterruptTime`

outFlags
A pointer to memory where return flags (see below) report on the block's actual memory location. See these constants:
`kQTSMemAllocAllocatedInTempMem`
`kQTSMemAllocAllocatedInSystemMem`

Return Value
A pointer to the newly allocated block.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSNewSourcer

Undocumented

```
OSErr QTSNewSourcer (
   void *params,
   const QTSSourcerInitParams *inInitParams,
   SInt32 inFlags,
   ComponentInstance *outSourcer
);
```

Parameters

params
Undocumented

inInitParams
Undocumented

inFlags
Undocumented

outSourcer
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSNewStatHelper

Creates a new statistics helper for a stream or presentation.

```
OSErr QTSNewStatHelper (
   QTSPresentation inPresentation,
   QTSStream inStream,
   OSType inStatType,
   SInt32 inFlags,
   QTSStatHelper *outStatHelper
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines the presentation to keep statistics on. To create a statistics helper for a particular stream, pass in `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines the stream to keep statistics on. To create a statistics helper for a whole presentation, pass in `kQTSAllStreams`.

inStatType

A constant (see below) that defines the type of statistic you want the statistics helper to gather. See these constants:

`kQTSAllStatisticsType`
`kQTSShortStatisticsType`
`kQTSSummaryStatisticsType`

inFlags

Constants (see below) governing the action of the statistics helper. See these constants:

`kQTSGetNameStatisticsFlag`
`kQTSDontGetDataStatisticsFlag`
`kQTSUpdateAtomsStatisticsFlag`
`kQTSGetUnitsStatisticsFlag`

outStatHelper

On entry, a pointer to a variable of type `QTSStatHelper`; on return, this variable is set to the new statistics helper.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

A statistics helper is a set of utility functions that you can use to retrieve and parse statistics from a stream component. You need to instantiate a statistics helper for every stream from which you want to gather statistics.

Special Considerations

When you are done using the statistics helper, call `QTSDisposeStatHelper` (page 18).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSNewStreamBuffer

Undocumented

```
OSErr QTSNewStreamBuffer (
   UInt32 inDataSize,
   SInt32 inFlags,
   QTSStreamBuffer **outStreamBuffer
);
```

Parameters

inDataSize
Undocumented

inFlags
Undocumented

outStreamBuffer
Undocumented

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSPrefsAddConnectionSetting

Undocumented

```
OSErr QTSPrefsAddConnectionSetting (
   OSType protocol,
   SInt32 portID,
   UInt32 flags,
   UInt32 seed
);
```

Parameters

protocol
A constant (see below) that identifies the connection protocol. See these constants:
kQTSDirectConnectHTTPProtocol
kQTSDirectConnectRTSPProtocol

portID
Undocumented

flags

Undocumented

seed

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPrefsAddProxySetting

Undocumented

```
OSErr QTSPrefsAddProxySetting (
   OSType proxyType,
   SInt32 portID,
   UInt32 flags,
   UInt32 seed,
   Str255 srvrURL
);
```

Parameters

proxyType

A constant (see below) that defines the proxy type. See these constants:

`kQTSHTTPProxyPrefsType`

`kQTSRTSPProxyPrefsType`

`kQTSSOCKSProxyPrefsType`

`kQTSDontProxyDataType`

portID

Undocumented

flags

Undocumented

seed

Undocumented

srvrURL

A string containing the server's URL.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPrefsAddProxyUserInfo

Undocumented

```
OSErr QTSPrefsAddProxyUserInfo (
   OSType proxyType,
   SInt32 flags,
   SInt32 flagsMask,
   StringPtr username,
   StringPtr password
);
```

Parameters

proxyType
Undocumented

flags
Undocumented

flagsMask
Undocumented

username
Undocumented

password
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.1 and later.

Declared In
`QuickTimeStreaming.h`

QTSPrefsFindConnectionByType

Undocumented

```
OSErr QTSPrefsFindConnectionByType (
   OSType protocol,
   UInt32 flags,
   UInt32 flagsMask,
   QTSTransportPref **connectionHndl,
   SInt16 *count
);
```

Parameters

protocol

A constant (see below) that identifies the connection protocol. See these constants:

`kQTSDirectConnectHTTPProtocol`

`kQTSDirectConnectRTSPProtocol`

flags

Undocumented

flagsMask

Undocumented

connectionHndl

A handle to a `QTSTransportPref` structure.

count

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPrefsFindProxyByType

Undocumented

```
OSErr QTSPrefsFindProxyByType (
   OSType proxyType,
   UInt32 flags,
   UInt32 flagsMask,
   QTSProxyPref **proxyHndl,
   SInt16 *count
);
```

Parameters

proxyType

A constant (see below) that defines the proxy type. See these constants:

`kQTSHTTPProxyPrefsType`

`kQTSRTSPProxyPrefsType`

`kQTSSOCKSProxyPrefsType`

`kQTSDontProxyDataType`

flags

Undocumented

flagsMask

Undocumented

proxyHndl

A handle to a `QTSProxyPref` structure.

count

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPrefsFindProxyUserInfoByType

Undocumented

```
OSErr QTSPrefsFindProxyUserInfoByType (
   OSType proxyType,
   SInt32 flags,
   SInt32 flagsMask,
   StringPtr username,
   StringPtr password
);
```

Parameters

proxyType

Undocumented

flags
Undocumented

flagsMask
Undocumented

username
Undocumented

password
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.1 and later.

Declared In
`QuickTimeStreaming.h`

QTSPrefsGetActiveConnection

Undocumented

```
OSErr QTSPrefsGetActiveConnection (
   OSType protocol,
   QTSTransportPref *connectInfo
);
```

Parameters

protocol
A constant (see below) that identifies the connection protocol. See these constants:
`kQTSDirectConnectHTTPProtocol`
`kQTSDirectConnectRTSPProtocol`

connectInfo
A pointer to a `QTSTransportPref` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPrefsGetInstantOnSettings

Undocumented

```
OSErr QTSPrefsGetInstantOnSettings (
   QTSInstantOnPref *outPref,
   SInt32 inFlags
);
```

Parameters

outPref

A pointer to a `QTSInstantOnPref` data structure.

inFlags

Undocumented

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`QuickTimeStreaming.h`

QTSPrefsGetNoProxyURLs

Undocumented

```
OSErr QTSPrefsGetNoProxyURLs (
   QTSNoProxyPref **noProxyHndl
);
```

Parameters

noProxyHndl

A handle to a `QTSNoProxyPref` structure.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPrefsSetInstantOnSettings

Undocumented

```
OSErr QTSPrefsSetInstantOnSettings (
   QTSInstantOnPref *inPref,
   SInt32 inFlags
);
```

Parameters

inPref

A pointer to a `QTSInstantOnPref` data structure.

inFlags

Undocumented

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`QuickTimeStreaming.h`

QTSPrefsSetNoProxyURLs

Undocumented

```
OSErr QTSPrefsSetNoProxyURLs (
   char *urls,
   UInt32 flags,
   UInt32 seed
);
```

Parameters

urls

A pointer to URL strings.

flags

Undocumented

seed

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresAddSourcer

Undocumented

```
OSErr QTSPresAddSourcer (
   QTSPresentation inPresentation,
   QTSStream inStream,
   ComponentInstance inSourcer,
   SInt32 inFlags
);
```

Parameters

inPresentation

Undocumented

inStream

Undocumented

inSourcer

Undocumented

inFlags

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresExport

Undocumented

```
OSErr QTSPresExport (
   QTSPresentation inPresentation,
   QTSStream inStream,
   QTSExportParams *inExportParams
);
```

Parameters

inPresentation

Undocumented

inStream

Undocumented

inExportParams

Undocumented

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetActiveSegment

Undocumented

```
OSErr QTSPresGetActiveSegment (
   QTSPresentation inPresentation,
   QTSStream inStream,
   TimeValue64 *outStartTime,
   TimeValue64 *outDuration
);
```

Parameters

inPresentation
A pointer to a `QTSPresentationRecord` structure.

inStream
A pointer to a `QTSStreamRecord` structure that defines a stream.

outStartTime
Undocumented

outDuration
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetClip

Gets the clipping region for a streaming presentation.

```
OSErr QTSPresGetClip (
   QTSPresentation inPresentation,
   QTSStream inStream,
   RgnHandle *outClip
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

outClip

A pointer to a handle to a `MacRegion` structure that defines a clipping region.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetDimensions

Gets the dimensions of a streaming presentation.

```
OSErr QTSPresGetDimensions (
   QTSPresentation inPresentation,
   QTSStream inStream,
   Fixed *outWidth,
   Fixed *outHeight
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

outWidth

A pointer to the width in pixels.

outHeight

A pointer to the height in pixels.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetEnable

Determines whether or not a presentation is enabled.

```
OSErr QTSPresGetEnable (
   QTSPresentation inPresentation,
   QTSStream inStream,
   Boolean *outEnableMode
);
```

Parameters

inPresentation
A pointer to a `QTSPresentationRecord` structure.

inStream
A pointer to a `QTSStreamRecord` structure that defines a stream.

outEnableMode
A pointer to a Boolean that is TRUE if the presentation is enabled, FALSE otherwise.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetFlags

Gets the flags currently set for a presentation.

```
OSErr QTSPresGetFlags (
   QTSPresentation inPresentation,
   SInt32 *outFlags
);
```

Parameters

inPresentation
A pointer to a `QTSPresentationRecord` structure that defines a presentation.

outFlags

On entry, the address of a variable of type SInt32; on return, this variable is set to the current flags (see below) for the specified presentation. See these constants:

kQTSAutoModeFlag

kQTSDontShowStatusFlag

kQTSSendMediaFlag

kQTSReceiveMediaFlag

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSPresGetGraphicsMode

Gets the graphics mode and blend color in use for video display by a stream or presentation.

```
OSErr QTSPresGetGraphicsMode (
   QTSPresentation inPresentation,
   QTSStream inStream,
   short *outMode,
   RGBColor *outOpColor
);
```

Parameters

inPresentation

A pointer to a QTSPresentationRecord structure that defines a presentation. If you want the graphics mode for a specific stream, pass the value kQTSInvalidPresentation.

inStream

A pointer to a QTSStreamRecord structure that defines a stream. If you want the graphics mode for the presentation as a whole, pass the value kQTSAllStreams.

outMode

On entry, a pointer to a short integer; on return, this variable is set to the graphics mode of the specified presentation or stream. See Graphics Transfer Modes.

outOpColor

On entry, the address of an RGBColor structure; on return, this structure is filled in with information about the color used for blending and transparent operations. The stream handler passes this color to QuickDraw as appropriate when you draw in addPin, subPin, blend, or transparent mode.

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetGWorld

Gets the graphics port and graphics device in use by a stream or presentation.

```
OSErr QTSPresGetGWorld (
   QTSPresentation inPresentation,
   QTSStream inStream,
   CGrafPtr *outGWorld,
   GDHandle *outGDHandle
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want the graphics mode for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want the graphics mode for the presentation as a whole, pass the value `kQTSAllStreams`.

outGWorld

On entry, the address of a variable of type `CGrafPtr`; on return, this variable is set to a pointer to a `CGrafPort` structure that defines the offscreen graphics world, color graphics port, or basic graphics port in use by the specified presentation or stream.

outGDHandle

On entry, the address of a variable of type `GDHandle`; on return, this variable is set to the handle of a `GDevice` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetIndSourcer

Undocumented

```
OSErr QTSPresGetIndSourcer (
   QTSPresentation inPresentation,
   QTSStream inStream,
   UInt32 inIndex,
   ComponentInstance *outSourcer
);
```

Parameters

inPresentation

Undocumented

inStream

Undocumented

inIndex

Undocumented

outSourcer

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresGetIndStream

Get a stream associated with a presentation, based on its index number.

```
QTSStream QTSPresGetIndStream (
   QTSPresentation inPresentation,
   UInt32 inIndex
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inIndex

The index number of the stream.

Return Value

A pointer to a `QTSStreamRecord` structure.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetInfo

Gets information about a presentation or stream.

```
OSErr QTSPresGetInfo (
   QTSPresentation inPresentation,
   QTSStream inStream,
   OSType inSelector,
   void *ioParam
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want information for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want information for the presentation as a whole, pass the value `kQTSAllStreams`.

inSelector

A constant (see below) that defines the information to be retrieved. See these constants:

- kQTSGetURLLink
- kQTSTargetBufferDurationInfo
- kQTSTargetBufferDurationInfo
- kQTSDurationInfo
- kQTSSourceTrackIDInfo
- kQTSSourceLayerInfo
- kQTSSourceLanguageInfo
- kQTSSourceTrackFlagsInfo
- kQTSSourceDimensionsInfo
- kQTSSourceVolumesInfo
- kQTSSourceMatrixInfo
- kQTSSourceClipRectInfo
- kQTSSourceGraphicsModeInfo
- kQTSSourceScaleInfo
- kQTSSourceBoundingRectInfo
- kQTSSourceUserDataInfo
- kQTSSourceInputMapInfo
- kQTSStatisticsInfo
- kQTSMinStatusDimensionsInfo
- kQTSNormalStatusDimensionsInfo
- kQTSTotalDataRateInfo
- kQTSTotalDataRateInInfo
- kQTSTotalDataRateOutInfo
- kQTSLostPercentInfo
- kQTSMediaTypeInfo
- kQTSNameInfo
- kQTSCanHandleSendDataType
- kQTSAnnotationsInfo

ioParam

A pointer to the retrieved information in the format shown below.

Return Value

See Error Codes. Returns noErr if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

QuickTimeStreaming.h

QTSPresGetMatrix

Gets the transformation matrix in use for the graphic display of a stream or presentation.

```
OSErr QTSPresGetMatrix (
   QTSPresentation inPresentation,
   QTSStream inStream,
   MatrixRecord *outMatrix
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to get the matrix for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to get the matrix for the presentation as a whole, pass the value `kQTSAllStreams`.

outMatrix

On entry, the address of a `MatrixRecord` structure; on return, this structure is filled with the transformation matrix in use by the stream handler. Note that the matrix passed back is the one last set by `QTSPresSetMatrix` (page 76), regardless of any additional matrixes that might have been used.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresGetNotificationProc

Gets the notification callback of a presentation.

```
OSErr QTSPresGetNotificationProc (
   QTSPresentation inPresentation,
   QTSNotificationUPP *outNotificationProc,
   void **outRefCon
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

outNotificationProc

A pointer to a Universal Procedure Pointer that accesses a `QTSNotificationProc` callback. The callback acts as a back channel from a presentation to its creator. The presentation sends notification of various events, such as a presentation, ending, or acknowledgment of a preroll request.

outRefCon
A handle to a constant to be passed to your `QTSNotificationProc`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetNumSourcers

Undocumented

```
UInt32 QTSPresGetNumSourcers (
   QTSPresentation inPresentation,
   QTSStream inStream
);
```

Parameters
inPresentation
Undocumented
inStream
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetNumStreams

Undocumented

```
UInt32 QTSPresGetNumStreams (
   QTSPresentation inPresentation
);
```

Parameters
inPresentation
A pointer to a `QTSPresentationRecord` structure.

Return Value
Undocumented

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetPicture

Undocumented

```
OSErr QTSPresGetPicture (
   QTSPresentation inPresentation,
   QTSStream inStream,
   PicHandle *outPicture
);
```

Parameters

inPresentation
A pointer to a `QTSPresentationRecord` structure.

inStream
A pointer to a `QTSStreamRecord` structure that defines a stream.

outPicture
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetPlayHints

Undocumented

```
OSErr QTSPresGetPlayHints (
   QTSPresentation inPresentation,
   QTSStream inStream,
   SInt32 *outFlags
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

outFlags

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetPreferredRate

Undocumented

```
OSErr QTSPresGetPreferredRate (
   QTSPresentation inPresentation,
   Fixed *outRate
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

outRate

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetPresenting

Determines whether presenting is enabled or disabled for a presentation or stream.

```
OSErr QTSPresGetPresenting (
   QTSPresentation inPresentation,
   QTSStream inStream,
   Boolean *outPresentingMode
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to get the presenting state for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to get the presenting state for the presentation as a whole, pass the value `kQTSAllStreams`.

outPresentingMode

A pointer to a Boolean that is TRUE if presenting is enabled, FALSE if it is disabled.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresGetSettings

Undocumented

```
OSErr QTSPresGetSettings (
   QTSPresentation inPresentation,
   QTSStream inStream,
   QTAtomContainer *outSettings,
   SInt32 inFlags
);
```

Parameters

inPresentation

Undocumented

inStream

Undocumented

outSettings

Undocumented

inFlags

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresGetSettingsAsText

Undocumented

```
OSErr QTSPresGetSettingsAsText (
   QTSPresentation inPresentation,
   QTSStream inStream,
   SInt32 inFlags,
   OSType inSettingsType,
   Handle *outText,
   QTSPanelFilterUPP inPanelFilterProc,
   void *inPanelFilterProcRefCon
);
```

Parameters

inPresentation
Undocumented

inStream
Undocumented

inFlags
Undocumented

inSettingsType
Undocumented

outText
Undocumented

inPanelFilterProc
Undocumented

inPanelFilterProcRefCon
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.1 and later.

Declared In
QuickTimeStreaming.h

QTSPresGetTimeBase

Undocumented

```
OSErr QTSPresGetTimeBase (
   QTSPresentation inPresentation,
   TimeBase *outTimeBase
);
```

Parameters

inPresentation

A pointer to a QTSPresentationRecord structure.

outTimeBase

Undocumented

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSPresGetTimeScale

Undocumented

```
OSErr QTSPresGetTimeScale (
   QTSPresentation inPresentation,
   TimeScale *outTimeScale
);
```

Parameters

inPresentation

A pointer to a QTSPresentationRecord structure.

outTimeScale

Undocumented

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSPresGetVolumes

Gets the sound volume levels of a stream or presentation.

```
OSErr QTSPresGetVolumes (
   QTSPresentation inPresentation,
   QTSStream inStream,
   short *outLeftVolume,
   short *outRightVolume
);
```

Parameters

inPresentation

A pointer to a QTSPresentationRecord structure that defines a presentation. If you want to get the volumes for a specific stream, pass the value kQTSInvalidPresentation.

inStream

A pointer to a QTSStreamRecord structure that defines a stream. If you want to get the volumes for the presentation as a whole, pass the value kQTSAllStreams.

outLeftVolume

On exit, the volume level of the left channel of the stream or presentation. The values returned may range from 0x0000 (silence) to 0x0100 (full volume).

outRightVolume

On exit, the volume level of the right channel of the stream or presentation. The values returned may range from 0x0000 (silence) to 0x0100 (full volume).

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeStreaming.h

QTSPresHasCharacteristic

Undocumented

```
OSErr QTSPresHasCharacteristic (
   QTSPresentation inPresentation,
   QTSStream inStream,
   OSType inCharacteristic,
   Boolean *outHasIt
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

inCharacteristic

Undocumented

outHasIt

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresIdle

Undocumented

```
void QTSPresIdle (
   QTSPresentation inPresentation,
   QTSPresIdleParams *ioParams
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

ioParams

Undocumented

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresInvalidateRegion

Undocumented

```
OSErr QTSPresInvalidateRegion (
   QTSPresentation inPresentation,
   RgnHandle inRegion
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inRegion

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresNewStream

Undocumented

```
OSErr QTSPresNewStream (
   QTSPresentation inPresentation,
   OSType inDataType,
   const void *inData,
   UInt32 inDataLength,
   SInt32 inFlags,
   QTSStream *outStream
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inDataType

Undocumented

inData

Undocumented

inDataLength

Undocumented

inFlags

Undocumented

outStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresPreroll

Undocumented

```
OSErr QTSPresPreroll (
   QTSPresentation inPresentation,
   QTSStream inStream,
   UInt32 inTimeValue,
   Fixed inRate,
   SInt32 inFlags
);
```

Parameters

inPresentation
A pointer to a `QTSPresentationRecord` structure.

inStream
A pointer to a `QTSStreamRecord` structure that defines a stream.

inTimeValue
Undocumented

inRate
Undocumented

inFlags
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresPreroll64

Undocumented

```
OSErr QTSPresPreroll64 (
   QTSPresentation inPresentation,
   QTSStream inStream,
   const TimeValue64 *inPrerollTime,
   Fixed inRate,
   SInt32 inFlags
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

inPrerollTime

Undocumented

inRate

Undocumented

inFlags

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresPreview

Undocumented

```
OSErr QTSPresPreview (
   QTSPresentation inPresentation,
   QTSStream inStream,
   const TimeValue64 *inTimeValue,
   Fixed inRate,
   SInt32 inFlags
);
```

Parameters

inPresentation

Undocumented

inStream

Undocumented

inTimeValue

Undocumented

inRate
Undocumented

inFlags
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresRemoveSourcer

Undocumented

```
OSErr QTSPresRemoveSourcer (
   QTSPresentation inPresentation,
   QTSStream inStream,
   ComponentInstance inSourcer,
   SInt32 inFlags
);
```

Parameters
inPresentation
Undocumented

inStream
Undocumented

inSourcer
Undocumented

inFlags
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetActiveSegment

Undocumented

```
OSErr QTSPresSetActiveSegment (
   QTSPresentation inPresentation,
   QTSStream inStream,
   const TimeValue64 *inStartTime,
   const TimeValue64 *inDuration
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

inStartTime

Undocumented

inDuration

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetClip

Undocumented

```
OSErr QTSPresSetClip (
   QTSPresentation inPresentation,
   QTSStream inStream,
   RgnHandle inClip
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

inClip

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetDimensions

Undocumented

```
OSErr QTSPresSetDimensions (
   QTSPresentation inPresentation,
   QTSStream inStream,
   Fixed inWidth,
   Fixed inHeight
);
```

Parameters

`inPresentation`
A pointer to a `QTSPresentationRecord` structure.

`inStream`
A pointer to a `QTSStreamRecord` structure that defines a stream.

`inWidth`
Undocumented

`inHeight`
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetEnable

Undocumented

```
OSErr QTSPresSetEnable (
   QTSPresentation inPresentation,
   QTSStream inStream,
   Boolean inEnableMode
);
```

Parameters

`inPresentation`
A pointer to a `QTSPresentationRecord` structure.

inStream
A pointer to a `QTSStreamRecord` structure that defines a stream.

inEnableMode
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetFlags

Undocumented

```
OSErr QTSPresSetFlags (
   QTSPresentation inPresentation,
   SInt32 inFlags,
   SInt32 inFlagsMask
);
```

Parameters

inPresentation
A pointer to a `QTSPresentationRecord` structure.

inFlags
Undocumented

inFlagsMask
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetGraphicsMode

Sets the graphics transfer mode for a streaming presentation.

```
OSErr QTSPresSetGraphicsMode (
   QTSPresentation inPresentation,
   QTSStream inStream,
   short inMode,
   const RGBColor *inOpColor
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

inMode

A short integer; see `Graphics Transfer Modes`.

inOpColor

A pointer to an `RGBColor` structure. This is the blend value for blends and the transparent color for transparent operations. The toolbox supplies this value to QuickDraw when you draw in `addPin`, `subPin`, `blend`, `transparent`, or `graphicsModeStraightAlphaBlend` mode.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresSetGWorld

Undocumented

```
OSErr QTSPresSetGWorld (
   QTSPresentation inPresentation,
   QTSStream inStream,
   CGrafPtr inGWorld,
   GDHandle inGDHandle
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream.

inGWorld

Undocumented

inGDHandle

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetInfo

Sets information for a presentation or stream.

```
OSErr QTSPresSetInfo (
   QTSPresentation inPresentation,
   QTSStream inStream,
   OSType inSelector,
   void *ioParam
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to set information for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to set information for the presentation as a whole, pass the value `kQTSAllStreams`.

inSelector

A constant (see below) that defines the type of information to be set. See these constants:

`kQTSGetURLLink`
`kQTSTargetBufferDurationInfo`
`kQTSDurationInfo`
`kQTSSourceTrackIDInfo`
`kQTSSourceLayerInfo`
`kQTSSourceLanguageInfo`
`kQTSSourceTrackFlagsInfo`
`kQTSSourceDimensionsInfo`
`kQTSSourceVolumesInfo`
`kQTSSourceMatrixInfo`
`kQTSSourceClipRectInfo`
`kQTSSourceGraphicsModeInfo`
`kQTSSourceScaleInfo`
`kQTSSourceBoundingRectInfo`
`kQTSSourceUserDataInfo`
`kQTSSourceInputMapInfo`

ioParam

A pointer to the information to be set in the format shown below.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetMatrix

Sets the transformation matrix to be used by the graphic display of a stream or presentation.

```
OSErr QTSPresSetMatrix (
   QTSPresentation inPresentation,
   QTSStream inStream,
   const MatrixRecord *inMatrix
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to set the matrix for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to set the matrix for the presentation as a whole, pass the value `kQTSAllStreams`.

inMatrix

A pointer to a `MatrixRecord` structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetNotificationProc

Sets the notification callback for a presentation.

```
OSErr QTSPresSetNotificationProc (
   QTSPresentation inPresentation,
   QTSNotificationUPP inNotificationProc,
   void *inRefCon
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure.

inNotificationProc

A Universal Procedure Pointer that accesses a `QTSNotificationProc` callback. The callback acts as a back channel from a presentation to its creator. The presentation sends notification of various events, such as a presentation, ending, or acknowledgment of a preroll request.

inRefCon

A pointer to data to be passed to your `QTSNotificationProc`.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSPresSetPlayHints

Undocumented

```
OSErr QTSPresSetPlayHints (
   QTSPresentation inPresentation,
   QTSStream inStream,
   SInt32 inFlags,
   SInt32 inFlagsMask
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to set the play hints for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to set the play hints for the presentation as a whole, pass the value `kQTSAllStreams`.

inFlags

Undocumented

inFlagsMask

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetPreferredRate

Undocumented

```
OSErr QTSPresSetPreferredRate (
   QTSPresentation inPresentation,
   Fixed inRate,
   SInt32 inFlags
);
```

Parameters

`inPresentation`
A pointer to a `QTSPresentationRecord` structure.

`inRate`
Undocumented

`inFlags`
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetPresenting

Enables or disables presentation of a stream to the user.

```
OSErr QTSPresSetPresenting (
   QTSPresentation inPresentation,
   QTSStream inStream,
   Boolean inPresentingMode
);
```

Parameters

`inPresentation`
A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to enable or disable the presentation for a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to enable or disable the presentation as a whole, pass the value `kQTSAllStreams`.

inPresentingMode

Pass TRUE to enable the presentation, FALSE to disable it.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetSettings

Undocumented

```
OSErr QTSPresSetSettings (
   QTSPresentation inPresentation,
   QTSStream inStream,
   QTAtomSpecPtr inSettings,
   SInt32 inFlags
);
```

Parameters

inPresentation

Undocumented

inStream

Undocumented

inSettings

Undocumented

inFlags

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSettingsDialog

Undocumented

```
OSErr QTSPresSettingsDialog (
   QTSPresentation inPresentation,
   QTSStream inStream,
   SInt32 inFlags,
   QTSModalFilterUPP inFilterProc,
   void *inFilterProcRefCon
);
```

Parameters

inPresentation
Undocumented

inStream
Undocumented

inFlags
Undocumented

inFilterProc
Undocumented

inFilterProcRefCon
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSettingsDialogWithFilters

Undocumented

```
OSErr QTSPresSettingsDialogWithFilters (
   QTSPresentation inPresentation,
   QTSStream inStream,
   SInt32 inFlags,
   QTSModalFilterUPP inFilterProc,
   void *inFilterProcRefCon,
   QTSPanelFilterUPP inPanelFilterProc,
   void *inPanelFilterProcRefCon
);
```

Parameters

inPresentation
Undocumented

inStream
Undocumented

inFlags
Undocumented

inFilterProc
Undocumented

inFilterProcRefCon
Undocumented

inPanelFilterProc
Undocumented

inPanelFilterProcRefCon
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.1 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSetVolumes

Sets the sound volume levels of a stream or presentation.

```
OSErr QTSPresSetVolumes (
   QTSPresentation inPresentation,
   QTSStream inStream,
   short inLeftVolume,
   short inRightVolume
);
```

Parameters

inPresentation
A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to set the volume of a specific stream, pass the value `kQTSInvalidPresentation`.

inStream
A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to set the volume of the presentation as a whole, pass the value `kQTSAllStreams`.

inLeftVolume
The volume level to be set for the left channel of the stream or presentation. The values may range from 0x0000 (silence) to 0x0100 (full volume).

inRightVolume
The volume level to be set for the right channel of the stream or presentation. The values may range from 0x0000 (silence) to 0x0100 (full volume).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSkipTo

Requests that a presentation skip to a given point, specified by a time value.

```
OSErr QTSPresSkipTo (
   QTSPresentation inPresentation,
   UInt32 inTimeValue
);
```

Parameters

`inPresentation`
A pointer to a `QTSPresentationRecord` structure.

`inTimeValue`
The time value to skip to, expressed in the time scale of the presentation.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresSkipTo64

Requests that a streaming presentation skip to a given point, specified by a 64-bit time value.

```
OSErr QTSPresSkipTo64 (
   QTSPresentation inPresentation,
   const TimeValue64 *inTimeValue
);
```

Parameters

`inPresentation`
A pointer to a `QTSPresentationRecord` structure.

inTimeValue

A pointer to a signed 64-bit integer that contains the time value to skip to, expressed in the time scale of the presentation.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresStart

Starts a streaming presentation or a stream.

```
OSErr QTSPresStart (
   QTSPresentation inPresentation,
   QTSStream inStream,
   SInt32 inFlags
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to start a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to start the presentation as a whole, pass the value `kQTSAllStreams`.

inFlags

Flags (see below) that govern the starting of the presentation or stream. See these constants:

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
If `QTSPresPreroll` (page 68) has not been called, QuickTime must set up the streams and do everything that would have been done in preroll. If the presentation has already been prerolled, it should be ready to start immediately.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresStop

Stops a streaming presentation or stream.

```
OSErr QTSPresStop (
   QTSPresentation inPresentation,
   QTSStream inStream,
   SInt32 inFlags
);
```

Parameters

inPresentation

A pointer to a `QTSPresentationRecord` structure that defines a presentation. If you want to stop a specific stream, pass the value `kQTSInvalidPresentation`.

inStream

A pointer to a `QTSStreamRecord` structure that defines a stream. If you want to stop the presentation as a whole, pass the value `kQTSAllStreams`. All audio and video output will cease.

inFlags

Flags that govern the stopping of the presentation or stream. No flags are currently defined.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSReleaseMemPtr

Disposes of a pointer to a streaming buffer that will be recirculated.

```
void QTSReleaseMemPtr (
   QTSMemPtr inMemPtr,
   SInt32 inFlags
);
```

Parameters

inMemPtr

A pointer to an opaque structure.

inFlags

Undocumented

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSSetNetworkAppName

Sets the name of a streaming network application.

```
OSErr QTSSetNetworkAppName (
   const char *inAppName,
   SInt32 inFlags
);
```

Parameters

inAppName

A pointer to a string containing the application's name.

inFlags

A flag (see below) that determines whether the name is a full pathname. See these constants:
`kQTSNetworkAppNameIsFullNameFlag`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSSourcerGetEnable

Undocumented

```
ComponentResult QTSSourcerGetEnable (
   QTSSourcer inSourcer,
   Boolean *outEnableMode,
   SInt32 inFlags
);
```

Parameters

inSourcer

Undocumented

outEnableMode

Undocumented

inFlags

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

QTSSourcerGetInfo

Undocumented

```
ComponentResult QTSSourcerGetInfo (
   QTSSourcer inSourcer,
   OSType inSelector,
   void *ioParams
);
```

Parameters

inSourcer
Undocumented

inSelector
Undocumented

ioParams
Undocumented

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

QTSSourcerGetTimeScale

Undocumented

```
ComponentResult QTSSourcerGetTimeScale (
   QTSSourcer inSourcer,
   TimeScale *outTimeScale
);
```

Parameters

inSourcer
Undocumented

outTimeScale
Undocumented

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSSourcerIdle

Undocumented

```
ComponentResult QTSSourcerIdle (
   QTSSourcer inSourcer,
   const TimeValue64 *inTime,
   SInt32 inFlags,
   SInt32 *outFlags
);
```

Parameters

`inSourcer`
Undocumented

`inTime`
Undocumented

`inFlags`
Undocumented

`outFlags`
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSSourcerInitialize

Undocumented

```
ComponentResult QTSSourcerInitialize (
   QTSSourcer inSourcer,
   const QTSSourcerInitParams *inInitParams
);
```

Parameters

`inSourcer`
Undocumented

inInitParams
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.1 and later.

Declared In
`QTStreamingComponents.h`

QTSSourcerSetEnable

Undocumented

```
ComponentResult QTSSourcerSetEnable (
   QTSSourcer inSourcer,
   Boolean inEnableMode,
   SInt32 inFlags
);
```

Parameters
inSourcer
Undocumented
inEnableMode
Undocumented
inFlags
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTSSourcerSetInfo

Undocumented

```
ComponentResult QTSSourcerSetInfo (
   QTSSourcer inSourcer,
   OSType inSelector,
   void *ioParams
);
```

Parameters

inSourcer

Undocumented

inSelector

Undocumented

ioParams

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

QTSSourcerSetTimeScale

Undocumented

```
ComponentResult QTSSourcerSetTimeScale (
   QTSSourcer inSourcer,
   TimeScale inTimeScale
);
```

Parameters

inSourcer

Undocumented

inTimeScale

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

QTSStatHelperGetNumStats

Gets the number of statistics that a statistic helper is reporting.

```
UInt32 QTSStatHelperGetNumStats (
   QTSStatHelper inStatHelper
);
```

Parameters

inStatHelper

A pointer to a `QTSStatHelperRecord` structure that defines the component instance of a statistics helper.

Return Value

The number of statistics.

Discussion

You can also find the number of statistics that a statistics helper is reporting by calling `QTSStatHelperResetIter` (page 91), then calling `QTSStatHelperNext` (page 91) iteratively until it returns FALSE and counting the iterations.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSStatHelperGetStats

Tells a statistics helper to update its statistics.

```
OSErr QTSStatHelperGetStats (
   QTSStatHelper inStatHelper
);
```

Parameters

inStatHelper

A pointer to a `QTSStatHelperRecord` structure that defines the component instance of a statistics helper.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

Statistics helpers update their statistics only when this function is called. You should call it at least once before calling `QTSStatHelperNext` (page 91), to ensure that the information returned is valid and current. The normal sequence is to call this function, then call `QTSStatHelperResetIter` (page 91), then make a series of calls to `QTSStatHelperNext` (page 91).

Version Notes

Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSStatHelperNext

Gets the next statistic from a statistic helper.

```
Boolean QTSStatHelperNext (
   QTSStatHelper inStatHelper,
   QTSStatHelperNextParams *ioParams
);
```

Parameters

inStatHelper

A pointer to a `QTSStatHelperRecord` structure that defines the component instance of a statistics helper.

ioParams

On entry, a pointer to a `QTSStatHelperNextParams` structure; on return, this structure is filled in with information about the next statistic from the specified statistic helper.

Return Value
FALSE if the last statistic has been returned, TRUE otherwise.

Discussion
You need to call this function once to retrieve each statistic. The normal sequence is to call `QTSStatHelperGetStats` (page 90), then call `QTSStatHelperResetIter` (page 91), then make a series of calls to this function until it returns FALSE.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSStatHelperResetIter

Reset the iteration counter of a statistics helper, so the next call to QTSStatHelperNext returns the first statistic.

```
OSErr QTSStatHelperResetIter (
   QTSStatHelper inStatHelper
);
```

Parameters

inStatHelper

A pointer to a `QTSStatHelperRecord` structure that defines the component instance of a statistics helper.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSStreamBufferDataInfo

Undocumented

```
void QTSStreamBufferDataInfo (
   QTSStreamBuffer *inStreamBuffer,
   unsigned char **outDataStart,
   UInt32 *outDataMaxLength
);
```

Parameters

`inStreamBuffer`
Undocumented

`outDataStart`
Undocumented

`outDataMaxLength`
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

RTPMPDoUserDialog

Obtains media-specific settings from the user through a dialog box.

```
ComponentResult RTPMPDoUserDialog (
   RTPMediaPacketizer rtpm,
   ModalFilterUPP inFilterUPP,
   Boolean *canceled
);
```

Parameters

rtpm

The component instance of the media packetizer.

inFilterUPP

A `ModalFilterProc` callback, which may be used in a call to the Mac OS `ModalDialog` function.

canceled

On return, a Boolean which is TRUE if the user pressed the cancel button in the dialog box. If this parameter is returned TRUE, the settings prior to calling this function should be retained.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function invokes a media packetizer's modal dialog to obtain user settings. If the packetizer supports "more settings," you can put up a dialog allowing the user to enter media-specific settings. You can determine whether a packetizer has this characteristic by calling `RTPMPHasCharacteristic` (page 100)). The settings can be obtained for storage by calling `RTPMPGetSettingsIntoAtomContainerAtAtom` (page 98), and can be restored or set directly from an application by calling `RTPMPSetSettingsFromAtomContainerAtAtom` (page 109).

Special Considerations

This function may be called at any time.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPMPFlush

Renamed RTPMPReset.

```
ComponentResult RTPMPFlush (
   RTPMediaPacketizer rtpm,
   SInt32 inFlags,
   SInt32 *outFlags
);
```

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPMPGetInfo

Obtains information of various types from a media packetizer.

```
ComponentResult RTPMPGetInfo (
   RTPMediaPacketizer rtpm,
   OSType inSelector,
   void *ioParams
);
```

Parameters

rtpm

The component instance of the media packetizer you want information from.

inSelector

The selector for the type information you want (see below). See these constants:

`kRTPMPPayloadTypeInfo`
`kRTPMPRTPTimeScaleInfo`
`kRTPMPRequiredSampleDescriptionInfo`
`kRTPMPMinPayloadSize`
`kRTPMPMinPacketDuration`
`kRTPMPSuggestedRepeatPktCountInfo`

ioParams

A pointer to a data structure of the appropriate type to hold the information you are requesting. You need to allocate and dispose of this data structure.

Return Value
See `Error Codes`. Returns `qtsBadSelectorErr` if `inSelector` requests a selector you do not support. Returns `noErr` if there is no error.

Discussion
This function can be called at any time.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming

qtstreaming.win

Declared In
QTStreamingComponents.h

RTPMPGetMaxPacketDuration

Reads the maximum packet duration currently set for this packetizer.

```
ComponentResult RTPMPGetMaxPacketDuration (
   RTPMediaPacketizer rtpm,
   UInt32 *outMaxPacketDuration
);
```

Parameters

rtpm

The component instance of the media packetizer.

outMaxPacketDuration

On return, a pointer to a 32-bit integer containing the maximum packet duration, in milliseconds, that the packetizer is set to use.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
The maximum allowable packet duration can change during a presentation, so you should obtain this value immediately before using it.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPMPGetMaxPacketSize

Returns the maximum packet size, in bytes, that the packetizer is set to create.

```
ComponentResult RTPMPGetMaxPacketSize (
   RTPMediaPacketizer rtpm,
   UInt32 *outMaxPacketSize
);
```

Parameters

rtpm

The component instance of the media packetizer.

outMaxPacketSize

On return, a pointer to a 32-bit integer containing the maximum packet size, in bytes, that the packetizer is set to create.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The maximum allowable packet size can change during a presentation, so you should obtain this value immediately before using it.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPGetMediaType

Obtains the data type being handled by a media packetizer.

```
ComponentResult RTPMPGetMediaType (
   RTPMediaPacketizer rtpm,
   OSType *outMediaType
);
```

Parameters

`rtpm`
The component instance of the media packetizer.

`outMediaType`
On return, a pointer to the media's data type, such as `VideoMediaType` or `SoundMediaType`; see `Data References`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

The media's data type must be set prior to calling `RTPMPSetSampleData` (page 107). It cannot change afterward.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPGetPacketBuilder

Obtains the component instance of the packet builder component being used by a media packetizer.

```
ComponentResult RTPMPGetPacketBuilder (
   RTPMediaPacketizer rtpm,
   ComponentInstance *outPacketBuilder
);
```

Parameters

rtpm

The component instance of the media packetizer whose packet builder you are interested in.

outPacketBuilder

On return, a pointer to the component instance of the packet builder component in use by this media packetizer.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPGetSettings

Undocumented

```
ComponentResult RTPMPGetSettings (
   RTPMediaPacketizer rtpm,
   QTAtomContainer *outSettings,
   SInt32 inFlags
);
```

Parameters

rtpm

Undocumented

outSettings

Undocumented

inFlags

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPGetSettingsAsText

Return the media-specific settings of a media packetizer as text in a format presentable to the user.

```
ComponentResult RTPMPGetSettingsAsText (
   RTPMediaPacketizer rtpm,
   Handle *text
);
```

Parameters

rtpm

The component instance of a media packetizer.

text

Return a handle to a copy of your user settings in text format. The text is formatted as simple array of characters. There is no size byte or null termination. Allocate the handle to fit the text precisely.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function expects you to return your user settings as text. It should be called only if the media packetizer supports packetizer-specific settings. To determine if your media packetizer supports this function, the application may call `RTPMPHasCharacteristic` (page 100).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPMPGetSettingsIntoAtomContainerAtAtom

Obtains the media-specific setting of a media packetizer.

```
ComponentResult RTPMPGetSettingsIntoAtomContainerAtAtom (
   RTPMediaPacketizer rtpm,
   QTAtomContainer inOutContainer,
   QTAtom inParentAtom
);
```

Parameters

rtpm

The component instance of the media packetizer.

inOutContainer

The atom container that holds the settings atom, which the caller must allocate.

inParentAtom

The atom that will hold the settings.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function should be called only if the media packetizer supports packetizer-specific settings. To determine if a media packetizer supports this function, call `RTPMPHasCharacteristic` (page 100).

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPGetTimeBase

Returns the time base passed to a media packetizer by RTPMPSetTimeBase.

```
ComponentResult RTPMPGetTimeBase (
   RTPMediaPacketizer rtpm,
   TimeBase *outTimeBase
);
```

Parameters

rtpm
The component instance of your media packetizer.

outTimeBase
A pointer to the time base passed to you by `RTPMPSetTimeBase` (page 110).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPGetTimeScale

Obtains the time scale in use by a media packetizer.

```
ComponentResult RTPMPGetTimeScale (
   RTPMediaPacketizer rtpm,
   TimeScale *outTimeScale
);
```

Parameters

rtpm
The component instance of media packetizer component.

outTimeScale

On return, contains a pointer to the time scale in use by the packetizer. The time scale indicates the number of time units that pass in one second when the media is playing at a rate of 1.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPHasCharacteristic

Determines whether a media packetizer has a particular characteristic, such as whether it supports a user settings dialog.

```
ComponentResult RTPMPHasCharacteristic (
   RTPMediaPacketizer rtpm,
   OSType inSelector,
   Boolean *outHasIt
);
```

Parameters

rtpm

The component instance of the media packetizer.

inSelector

A selector for the characteristic you want to know about. See these constants:

`kRTPMPNoSampleDataRequiredCharacteristic`

`kRTPMPHasUserSettingsDialogCharacteristic`

`kRTPMPPrefersReliableTransportCharacteristic`

`kRTPMPRequiresOutOfBandDimensionsCharacteristic`

outHasIt

On return, contains a Boolean value that is TRUE if the media packetizer has this characteristic, FALSE otherwise.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming

qtstreaming.win

Declared In
QTStreamingComponents.h

RTPMPIdle

Called periodically in your event loop to allocate time to each media packetizer.

```
ComponentResult RTPMPIdle (
   RTPMediaPacketizer rtpm,
   SInt32 inFlags,
   SInt32 *outFlags
);
```

Parameters

rtpm
The component instance of the media packetizer.

inFlags
There are currently no defined flags.

outFlags
On return, contains a pointer to a signed 32-bit integer that holds a flag (see below) from the packetizer. See these constants:
kRTPMPStillProcessingData

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
The packetizer will use this time to process the data in its buffer. If the data has not all been processed, this function returns the kRTPMPStillProcessingData flag. Data is placed in the buffer by RTPMPSetSampleData (page 107).

Special Considerations

The packetizer may make calls to the packet builder in response to this call.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPMPInitialize

Initializes a media packetizer component.

```
ComponentResult RTPMPInitialize (
   RTPMediaPacketizer rtpm,
   SInt32 inFlags
);
```

Parameters

rtpm

The component instance of the media packetizer.

inFlags

A signed 32-bit integer containing the flags (see below) you wish to pass to the packetizer at start-up. See these constants:

`kRTPMPRealtimeModeFlag`

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The calling component must call this function before sending any data to a media packetizer or making any `RTPMPSet` calls. The calling component then calls `RTPMPSetSampleData` (page 107) and `RTPMPIdle` (page 101) repeatedly. The calling component passes sample data (obtained, for example, from `GetMediaSample`), to the media packetizer by calling `RTPMPSetSampleData`. If `RTPMPSetSampleData` or `RTPMPIdle` return the flag `kRTPMPStillProcessingData`, then the calling component should call `RTPMPIdle`; if not, it is free to call `RTPMPSetSampleData` again.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTSPketizerReassem

QTSPketizerReassem.win

qtstreaming

qtstreaming.win

Declared In

`QTStreamingComponents.h`

RTPMPPreflightMedia

Determines whether your packetizer can work with a given media type and sample description.

```
ComponentResult RTPMPPreflightMedia (
   RTPMediaPacketizer rtpm,
   OSType inMediaType,
   SampleDescriptionHandle inSampleDescription
);
```

Parameters

rtpm

The component instance of your media packetizer.

inMediaType
The media type, such as `'vide'`; see `Data References`.

inSampleDescription
A handle to the `SampleDescription` structure.

Return Value
Return `noErr` if you can packetize this type of data; return `qtsUnsupportedFeatureErr` if you cannot. See `Error Codes`.

Discussion
This function must be implemented by your packetizer. It will be called before you are asked to packetize any data.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPReset

Allows a media packetizer to stop packetizing its current input, set its state to idle, and flush its input buffer.

```
ComponentResult RTPMPReset (
   RTPMediaPacketizer rtpm,
   SInt32 inFlags
);
```

Parameters
rtpm
The component instance of the media packetizer.

inFlags
A signed 32-bit integer containing any flags you are passing to the media packetizer. There are currently no defined flags.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
You can use this function to stop the media packetizer and flush its input buffer when you wish to stop transmitting immediately, when you are skipping forward or backward in the stream, or if the network data connection is interrupted.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win

qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPMPSetInfo

Sets any one of several parameters for a media packetizer.

```
ComponentResult RTPMPSetInfo (
   RTPMediaPacketizer rtpm,
   OSType inSelector,
   const void *ioParams
);
```

Parameters

rtpm

The component instance of the media packetizer.

inSelector

A selector (see below) for the type of information you wish to set. See these constants:

`kQTSSourceTrackIDInfo`
`kQTSSourceLayerInfo`
`kQTSSourceLanguageInfo`
`kQTSSourceTrackFlagsInfo`
`kQTSSourceDimensionsInfo`
`kQTSSourceVolumesInfo`
`kQTSSourceMatrixInfo`
`kQTSSourceClipRectInfo`
`kQTSSourceGraphicsModeInfo`
`kQTSSourceBoundingRectInfo`
`kQTSSourceScaleInfo`
`kQTSSourceUserDataInfo`
`kQTSSourceInputMapInfo`

ioParams

A pointer to a data structure of the appropriate type for the information you are passing.

Return Value
Return `qtsBadSelectorErr` if you do not support the selector. Return `noErr` if there is no error. See `Error Codes`.

Discussion
This function is used to pass track-level information about the media track to be packetized, such as its track ID, layer, and transformation matrix. Return `qtsBadSelectorErr` unless your packetizer is able to transmit this kind of data to your reassembler for use in the client movie.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPSetMaxPacketDuration

Sets the maximum packet duration that the media packetizer is to use.

```
ComponentResult RTPMPSetMaxPacketDuration (
   RTPMediaPacketizer rtpm,
   UInt32 inMaxPacketDuration
);
```

Parameters

rtpm

The component instance of the media packetizer.

inMaxPacketDuration

An unsigned 32-bit integer containing the maximum packet duration in milliseconds. This value should not be smaller than the value returned from `RTPMPGetInfo` (page 94) with the `kRTPMPMinPacketDuration` selector.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The maximum packet duration cannot be changed during a presentation, and this function cannot be called after calling `RTPMPSetSampleData` (page 107).

Special Considerations

If `RTPMPSetMaxPacketDuration` is not called, a default value will be used.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPSetMaxPacketSize

Sets the maximum packet size for packets created by a media packetizer.

```
ComponentResult RTPMPSetMaxPacketSize (
   RTPMediaPacketizer rtpm,
   UInt32 inMaxPacketSize
);
```

Parameters

rtpm

The component instance of the media packetizer.

inMaxPacketSize

An unsigned 32-bit integer specifying the maximum size, in bytes, of packets to be created. This value must not be smaller than the value returned from `RTPMPGetInfo` (page 94) with the `kRTPMPMinPayloadSize` selector. The media packetizer will not create packets larger than this value. The limit applies only to the payload data.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The maximum packet size cannot change during a presentation. Streaming will be most efficient if this value is set to the largest packet size that can traverse the network without being split. `RTPMPSetMaxPacketSize` may not be called after calling `RTPMPSetSampleData` (page 107).

Special Considerations

If `RTPMPSetMaxPacketSize` is not called, a default value will be used.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPMPSetMediaType

Sets the type of media that a media packetizer will process.

```
ComponentResult RTPMPSetMediaType (
   RTPMediaPacketizer rtpm,
   OSType inMediaType
);
```

Parameters

rtpm

The component instance of the media packetizer.

inMediaType

The media type; see `Data References`.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The media type must be set prior to calling `RTPMPSetSampleData` (page 107) and cannot change after such calls.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPSetPacketBuilder

Selects which packet builder a media packetizer will use.

```
ComponentResult RTPMPSetPacketBuilder (
   RTPMediaPacketizer rtpm,
   ComponentInstance inPacketBuilder
);
```

Parameters

rtpm
The component instance of the media packetizer.

inPacketBuilder
The component instance of the packet builder component to use.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
A media packetizer always sends its output to a packet builder. The specified packet builder may assemble actual RTP packets, or it may use information about the packet to build a hint track. You must set the packet builder using this call prior to any calls to `RTPMPSetSampleData` (page 107). You can also use this function to dynamically change the packet builder a media packetizer uses.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPSetSampleData

Provides sample data directly to a media packetizer component.

```
ComponentResult RTPMPSetSampleData (
   RTPMediaPacketizer rtpm,
   const RTPMPSampleDataParams *inSampleData,
   SInt32 *outFlags
);
```

Parameters

rtpm

The component instance of the media packetizer.

inSampleData

A pointer to a `RTPMPSampleDataParams` structure containing the sample data you are passing. Calling this routine adds data cumulatively to any previous calls to this function. The data can contain any number of samples (1 or more), or a partial sample.

outFlags

Flags (see below) that indicate processing status. This function will return `kRTPMPWantsMoreDataFlag` if it has completed processing of all pending data. Otherwise, you must make calls to `RTPMPIdle` (page 101) until this function no longer returns `kRTPMPStillProcessingData`. See these constants:

`kRTPMPStillProcessingData`

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This routine is called to pass media data directly to a media packetizer. The packetizer will not copy this data; it will call the release callback when it is finished with it. The media packetizer may or may not make calls to the packet builder in response to this call.

Special Considerations

This call is normally followed by a series of calls to `RTPMPIdle` (page 101), which grants time to the media packetizer in order to process the data passed by this function.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPMPSetSettings

Undocumented

```
ComponentResult RTPMPSetSettings (
   RTPMediaPacketizer rtpm,
   QTAtomSpecPtr inSettings,
   SInt32 inFlags
);
```

Parameters

rtpm

Undocumented

inSettings
Undocumented

inFlags
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPSetSettingsFromAtomContainerAtAtom

Sets the media-specific settings of a media packetizer, using an atom inside an atom container.

```
ComponentResult RTPMPSetSettingsFromAtomContainerAtAtom (
   RTPMediaPacketizer rtpm,
   QTAtomContainer inContainer,
   QTAtom inParentAtom
);
```

Parameters

rtpm
The component instance of the media packetizer.

inContainer
The atom container that holds the settings atom.

inParentAtom
The atom that holds the settings.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function should be called only if the media packetizer supports packetizer-specific settings. To determine if a media packetizer supports this function, call `RTPMPHasCharacteristic` (page 100).

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPSetTimeBase

Tells your packetizer what time base is in use by the calling application.

```
ComponentResult RTPMPSetTimeBase (
   RTPMediaPacketizer rtpm,
   TimeBase inTimeBase
);
```

Parameters

rtpm

Component instance of your packetizer.

inTimeBase

The time base in use for this stream. You can query this time base to find out the current time in the stream.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function may be called during setup for a live transmission.

Special Considerations

Your packetizer should not rely on receiving this call.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPMPSetTimeScale

Sets the time scale the media packetizer will use.

```
ComponentResult RTPMPSetTimeScale (
   RTPMediaPacketizer rtpm,
   TimeScale inTimeScale
);
```

Parameters

rtpm

The component instance of the media packetizer.

inTimeScale

The time scale to use.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The time scale is the number of time units that pass in one second when the media is playing at a rate of 1. This time scale gives meaning to the times used when calling `RTPMPSetSampleData` (page 107).

Special Considerations

The time scale must be set before calling `RTPMPSetSampleData` (page 107).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPPBAddPacketLiteralData

Passes literal data directly to a packet builder component.

```
ComponentResult RTPPBAddPacketLiteralData (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   UInt8 *inData,
   UInt32 inDataLength,
   RTPPacketRepeatedDataRef *outDataRef
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket

The RTP packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

inData

A pointer to the data you are passing.

inDataLength

An unsigned 32-bit integer containing the length, in bytes, of the data you are passing.

outDataRef

On return, contains a pointer to a data reference. Use this reference if you wish to later tell the packet builder to use this same data again, without having to literally pass the data again. Pass in `NIL` if you do not need the packet builder to repeat the data. If you do not pass in `NIL`, you must dispose of the data explicitly by calling `RTPPBReleaseRepeatedData` (page 124).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function will return a reference which can be used to specify the same data repeatedly without having to pass in the data again. This is done by calling `RTPPBAddPacketRepeatedData` (page 112) with the reference which was returned by this function. For example, you can use this function to insert static header information into a packet prior to inserting media sample data. It will return a data reference you can use to insert the same static information into later packets.

Special Considerations

To specify media data to be placed in a packet, a media packetizer should call `RTPPBAddPacketSampleData` (page 113).

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPPBAddPacketRepeatedData

Tells a packet builder component to insert previously-specified data into a packet.

```
ComponentResult RTPPBAddPacketRepeatedData (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   RTPPacketRepeatedDataRef inDataRef
);
```

Parameters

rtpb
The component instance of the packet builder component.

inFlags
A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup
The packet group containing the packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket
The RTP packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

inDataRef

A reference to the data to repeat. This is normally a data reference returned by RTPPBAddPacketLiteralData (page 111) or RTPPBAddPacketSampleData (page 113).

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
Use this function to cause a packet builder component to repeatedly insert the same data into packets without having to pass the data each time. This is typically done to repeat static header information into a series of packets, or to insert previously-sent sample data into a redundant packet. The data is first specified by a call to RTPPBAddPacketLiteralData (page 111) or RTPPBAddPacketSampleData (page 113), which inserts the data the first time and returns a data reference. The data reference is then used with this function to send the data again.

Special Considerations

When you are done sending the repeated data, release the data structure by calling RTPPBReleaseRepeatedData (page 124).

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPPBAddPacketSampleData

Commands a packet builder component to insert media sample data into a packet.

```
ComponentResult RTPPBAddPacketSampleData (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   RTPMPSampleDataParams *inSampleDataParams,
   UInt32 inSampleOffset,
   UInt32 inSampleDataLength,
   RTPPacketRepeatedDataRef *outDataRef
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the packet into which the data will be placed. This is normally a reference returned by RTPPBBeginPacketGroup (page 117).

inPacket

The RTP packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

inSampleDataParams

A pointer to a `RTPMPSampleDataParams` structure for the sample data you are inserting.

inSampleOffset

A 32-bit unsigned integer containing the offset into the sample media, in bytes.

inSampleDataLength

A 32-bit unsigned integer specifying the number of bytes of media sample data to insert into the packet.

outDataRef

On return, contains a pointer to a data reference. Use this reference if you wish to later tell the packet builder to use this same sample data again, without having to literally pass the data again. Pass in `NIL` if you do not need the packet builder to repeat the data. If you do not pass in `NIL`, you must dispose of the data explicitly by calling `RTPPBReleaseRepeatedData` (page 124).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function will return a reference which can be used to specify the same data repeatedly without having to pass in the data again. The media packetizer specifies the offset into the media and the length of the sample to insert. You can insert data repeatedly by calling `RTPPBAddPacketRepeatedData` (page 112) with the reference which was returned by `RTPPBAddPacketLiteralData` (page 111).

Special Considerations

When a reference is no longer needed, it should be disposed of by using the call `RTPPBReleaseRepeatedData` (page 124).

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPPBAddPacketSampleData64

Provides a 64-bit version of RTPPBAddPacketSampleData for large sample media.

```
ComponentResult RTPPBAddPacketSampleData64 (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   RTPMPSampleDataParams *inSampleDataParams,
   const UInt64 *inSampleOffset,
   UInt32 inSampleDataLength,
   RTPPacketRepeatedDataRef *outDataRef
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket

The RTP packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

inSampleDataParams

A pointer to a `RTPMPSampleDataParams` structure for the sample data you are inserting.

inSampleOffset

A 64-bit unsigned integer containing the offset into the sample media, in bytes.

inSampleDataLength

A 32-bit unsigned integer specifying the number of bytes of media sample data to insert into the packet.

outDataRef

On return, contains a pointer to a data reference. Use this reference if you wish to later tell the packet builder to use this same sample data again, without having to literally pass the data again. Pass in `NIL` if you do not need the packet builder to repeat the data. If you do not pass in `NIL`, you must dispose of the data explicitly by calling `RTPPBReleaseRepeatedData` (page 124).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPPBAddRepeatPacket

Undocumented

```
ComponentResult RTPPBAddRepeatPacket (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   TimeValue inTransmissionOffset,
   UInt32 inSequenceNumber
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket

The RTP packet into which the data will be placed. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

inTransmissionOffset

Undocumented

inSequenceNumber

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPPBBeginPacket

Tells a packet builder to create a new packet.

```
ComponentResult RTPPBBeginPacket (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   UInt32 inPacketMediaDataLength,
   RTPPacketRef *outPacket
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the new packet. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacketMediaDataLength

An unsigned 32-bit integer specifying the maximum length of data that will be inserted into this packet. This includes the data for all subsequent `RTPPBAddPacketLiteralData` (page 111), `RTPPBAddPacketSampleData` (page 113), and `RTPPBAddPacketRepeatedData` (page 112) calls until the packet is closed. The value of this parameter may be larger, but must not be smaller, than the amount of data inserted in the packet.

outPacket

On return, contains a pointer to the packet. Use this reference to insert data into the packet.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The media packetizer uses this function to create each new packet, before inserting any literal, repeated, or sample data. A call to `RTPPBBeginPacketGroup` (page 117) must be made before creating the first packet in a group. Data can be inserted into the packet using `RTPPBAddPacketLiteralData` (page 111), `RTPPBAddPacketRepeatedData` (page 112), or `RTPPBAddPacketSampleData` (page 113). When the packet is complete, call `RTPPBEndPacket` (page 118).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTSPketizerReassem

QTSPketizerReassem.win

qtstreaming

qtstreaming.win

Declared In

`QTStreamingComponents.h`

RTPPBBeginPacketGroup

Tells a packet builder to create a new packet group.

```
ComponentResult RTPPBBeginPacketGroup (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   UInt32 inTimeStamp,
   RTPPacketGroupRef *outPacketGroup
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inTimeStamp

A unsigned 32-bit integer containing the time stamp for this packet group.

outPacketGroup

On return, contains a pointer to a reference to the packet group. Use this data reference when creating a new packet or inserting data into a packet that belongs to this group.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

A media packetizer creates a packet group using this function. The data reference returned by this function is then used to create a series of packets that belong to this group. The data reference is also required when inserting data into packets.

Special Considerations

When the packet group is complete, call `RTPPBEndPacketGroup` (page 119).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTSPketizerReassem

QTSPketizerReassem.win

qtstreaming

qtstreaming.win

Declared In

`QTStreamingComponents.h`

RTPPBEndPacket

Tells a packet builder that a packet is complete.

```
ComponentResult RTPPBEndPacket (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   UInt32 inTransmissionTimeOffset,
   UInt32 inDuration
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the new packet. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket

The RTP packet containing the data. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

inTransmissionTimeOffset

The time offset at which the media sample data contained in this packet begins, in milliseconds. This offset is added to the RTP transmission time to determine when to send the packet.

inDuration

The duration of this packet, specified in milliseconds.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Call this function once when each packet is complete.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPPBEndPacketGroup

Tells a packet builder component that a packet group is complete.

```
ComponentResult RTPPBEndPacketGroup (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

A data reference to the packet group being ended. This is normally a data reference returned by `RTPPBBeginPacketGroup` (page 117).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function should be called when all the packets in a group are complete and the media packetizer is ready either to create a new packet group or to terminate the stream.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTSPketizerReassem

QTSPketizerReassem.win

qtstreaming

qtstreaming.win

Declared In

`QTStreamingComponents.h`

RTPPBGetCallback

Gets the callback used to communicate with the caller of a media packetizer.

```
ComponentResult RTPPBGetCallback (
   RTPPacketBuilder rtpb,
   RTPPBCallbackUPP *outCallback,
   void **outRefCon
);
```

Parameters

rtpb

The component instance of the packet builder component.

outCallback

A pointer to an `RTPPBCallbackProc` callback.

outRefCon
A handle to any data your callback needs.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPPBGetInfo

Gets information about a streaming packet builder.

```
ComponentResult RTPPBGetInfo (
   RTPPacketBuilder rtpb,
   OSType inSelector,
   void *ioParams
);
```

Parameters
rtpb
The component instance of the packet builder component.

inSelector
A constant (see below) that defines the type of information to retrieve. See these constants:

ioParams
A pointer to the retrieved information.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPPBGetPacketSequenceNumber

Gets the relative sequence number for a streaming packet.

```
ComponentResult RTPPBGetPacketSequenceNumber (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   UInt32 *outSequenceNumber
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

Undocumented

inPacketGroup

A data reference to a packet group. This is normally a data reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket

The RTP packet. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

outSequenceNumber

A pointer to the sequence number.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPPBGetPacketTimeStampOffset

Undocumented

```
ComponentResult RTPPBGetPacketTimeStampOffset (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   SInt32 *outTimeStampOffset
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the packet of interest. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket

The RTP packet of interest. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

outTimeStampOffset

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPPBGetSampleData

Undocumented

```
ComponentResult RTPPBGetSampleData (
   RTPPacketBuilder rtpb,
   RTPMPSampleDataParams *inParams,
   const UInt64 *inStartOffset,
   UInt8 *outDataBuffer,
   UInt32 inBytesToRead,
   UInt32 *outBytesRead,
   SInt32 *outFlags
);
```

Parameters

rtpb

The component instance of the packet builder component.

inParams

A pointer to a `RTPMPSampleDataParams` structure.

inStartOffset

Undocumented

outDataBuffer

Undocumented

inBytesToRead

Undocumented

outBytesRead

Undocumented

outFlags

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPPBReleaseRepeatedData

Lets a packet builder deallocate data that will no longer be used.

```
ComponentResult RTPPBReleaseRepeatedData (
   RTPPacketBuilder rtpb,
   RTPPacketRepeatedDataRef inDataRef
);
```

Parameters

rtpb

The component instance of the packet builder component.

inDataRef

The data reference to the repeated data. This is normally a data reference returned by `RTPPBAddPacketLiteralData` (page 111) or `RTPPBAddPacketSampleData` (page 113).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
You must release the data if you have allowed `RTPPBAddPacketLiteralData` (page 111) or `RTPPBAddPacketSampleData` (page 113) to return a data reference, even if you have not called `RTPPBAddPacketRepeatedData` (page 112).You must either pass `NIL` to the data reference when adding literal or sample data, or you must release the data by calling this function.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPPBSetCallback

Sets the callback used to communicate with the caller of a media packetizer.

```
ComponentResult RTPPBSetCallback (
   RTPPacketBuilder rtpb,
   RTPPBCallbackUPP inCallback,
   void *inRefCon
);
```

Parameters

rtpb

The component instance of the packet builder component.

inCallback

A Universal Procedure Pointer that references an `RTPPBCallbackProc` callback.

inRefCon

A pointer to any data your callback needs.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPPBSetInfo

Sets information for a streaming packet builder.

```
ComponentResult RTPPBSetInfo (
   RTPPacketBuilder rtpb,
   OSType inSelector,
   void *ioParams
);
```

Parameters

rtpb

The component instance of the packet builder component.

inSelector

A constant (see below) that defines the type of information to set. See these constants:

ioParams

A pointer to the information.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPPBSetPacketSequenceNumber

Sets the relative sequence number for a streaming packet.

```
ComponentResult RTPPBSetPacketSequenceNumber (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   UInt32 inSequenceNumber
);
```

Parameters

rtpb
The component instance of the packet builder component.

inFlags
Undocumented

inPacketGroup
A data reference to a packet group. This is normally a data reference returned by RTPPBBeginPacketGroup (page 117).

inPacket
The RTP packet. This is normally a reference returned by RTPPBBeginPacket (page 116).

inSequenceNumber
The sequence number to be set.

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPPBSetPacketTimeStampOffset

Undocumented

```
ComponentResult RTPPBSetPacketTimeStampOffset (
   RTPPacketBuilder rtpb,
   SInt32 inFlags,
   RTPPacketGroupRef inPacketGroup,
   RTPPacketRef inPacket,
   SInt32 inTimeStampOffset
);
```

Parameters

rtpb

The component instance of the packet builder component.

inFlags

A signed 32-bit integer containing any flags you are passing. There are currently no defined flags.

inPacketGroup

The packet group containing the packet of interest. This is normally a reference returned by `RTPPBBeginPacketGroup` (page 117).

inPacket

The RTP packet of interest. This is normally a reference returned by `RTPPBBeginPacket` (page 116).

inTimeStampOffset

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPRssmAdjustPacketParams

Called by the base reassembler when it is processing a packet, allowing your packet reassembler to adjust the packet parameters before the packet is processed.

```
ComponentResult RTPRssmAdjustPacketParams (
   RTPReassembler rtpr,
   RTPRssmPacket *inPacket,
   SInt32 inFlags
);
```

Parameters

rtpr

The component instance of your packet reassembler

inPacket

A pointer to the `packet` whose parameters can be adjusted.

inFlags

A signed 32-bit integer containing any flags (see below) being passed to your packet reassembler.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Your packet reassembler can adjust the following parameters in each packet: `payloadHeaderLength`, `dataLength`, `serverEditParams`, and `chunkFlags`. If your packet reassembler does not implement this function, or takes no action, the default for these parameters will be: `payloadHeaderLength` =fixed header length that is set (default is 0); `dataLength` =`packetData` - `transportHeaderLength` - `payloadHeaderLength`; no `serverEditParams`; `chunkFlags` =0.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmClearCachedPackets

Forces the base reassembler to flush all packets currently queued in its lists.

```
ComponentResult RTPRssmClearCachedPackets (
   RTPReassembler rtpr,
   SInt32 inFlags
);
```

Parameters

`rtpr`
The component instance of the base reassembler.

`inFlags`
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function retains the last sequence number and related information. It is useful only when the base reassembler is operating with the `kRTPRssmQueueAndUseMarkerBitFlag` flag set; see `RTPRssmSetCapabilities` (page 146).

Version Notes
Introduced in QuickTime 4.1. Replaces `RTPRssmFlushPackets`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmComputeChunkSize

Lets your packet reassembler compute the size of a chunk, based on the packet list for the chunk, using your own algorithm.

```
ComponentResult RTPRssmComputeChunkSize (
   RTPReassembler rtpr,
   RTPRssmPacket *inPacketListHead,
   SInt32 inFlags,
   UInt32 *outChunkDataSize
);
```

Parameters

rtpr

The component instance of your packet reassembler.

inPacketListHead

A pointer to the list of packets that make up this chunk.

inFlags

A signed 32-bit integer containing any flags being passed to your packet reassembler.

outChunkDataSize

You should return a pointer to an unsigned 32-bit variable containing the calculated size for this chunk.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function is called once for each packet list. Implement this function only if you need to override the base reassembler's default computation. If you do not implement this call, the base reassembler will compute the chunk size by summing the data lengths for all packets in the list.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmCopyDataToChunk

Lets your packet reassembler write the chunk data, based on the list of packets for the chunk, using your own algorithm.

```
ComponentResult RTPRssmCopyDataToChunk (
   RTPReassembler rtpr,
   RTPRssmPacket *inPacketListHead,
   UInt32 inMaxChunkDataSize,
   SHChunkRecord *inChunk,
   SInt32 inFlags
);
```

Parameters

rtpr

The component instance of your packet reassembler.

inPacketListHead

A pointer to the list of packets that make up this chunk.

inMaxChunkDataSize

An unsigned 32-bit integer containing the maximum allowable chunks size.

inChunk

A pointer to the chunk record. Write the chunk data to this record.

inFlags

A 32-bit signed integer containing any flags being passed to your media packetizer.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function is useful, for example, when an H.261 packet reassembler must adjust the byte at packet boundaries. Implement this function only if you need to override the base reassembler's default behavior. If you do not implement this function, the base reassembler will write the chunk data by taking `dataLength` bytes from each packet, starting at an offset of (`packetData` + `transportHeaderLength` + `payloadHeaderLength`).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPRssmDecrChunkRefCount

Tells the base reassembler to dispose of a chunk that it has created or preserved for you.

```
ComponentResult RTPRssmDecrChunkRefCount (
   RTPReassembler rtpr,
   SHChunkRecord *inChunk
);
```

Parameters

rtpr

The component instance of the base reassembler component.

inChunk

A pointer to the chunk record to dispose.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
If you have overridden `RTPRssmSendPacketList` (page 144) behavior, and are instructing the base reassembler to construct chunks manually, your packet assembler must explicitly dispose of the chunks by calling either this function or `RTPRssmSendChunkAndDecrRefCount` (page 143). This function is also used to release a chunk you have preserved using `RTPRssmIncrChunkRefCount` (page 139).

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPRssmFillPacketListParams

Fills in a packet structure manually.

```
ComponentResult RTPRssmFillPacketListParams (
   RTPReassembler rtpr,
   RTPRssmPacket *inPacketListHead,
   SInt32 inNumWraparounds,
   SInt32 inFlags
);
```

Parameters

rtpr
The component instance of the base reassembler.

inPacketListHead
A pointer to the `RTPRssmPacket` packet structure.

inNumWraparounds
The high-order 32 bits of the timestamp for this packet. The low-order 32 bits are found in the RTP packet header.

inFlags
A signed 32-bit integer containing any flags you are passing to the base reassembler.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Call this function only if your packet reassembler is overriding the `RTPRssmSendPacketList` (page 144) behavior. The base reassembler will call back to your packet reassembler using `RTPRssmAdjustPacketParams` (page 127) and `RTPRssmComputeChunkSize` (page 129).

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmGetCapabilities

Obtains the current flag settings for the base reassembler.

```
ComponentResult RTPRssmGetCapabilities (
   RTPReassembler rtpr,
   SInt32 *outFlags
);
```

Parameters

rtpr

The component instance of the base reassembler.

outFlags

On return, contains a pointer to the reassembler's current flags (see below). See these constants:

`kRTPRssmEveryPacketAChunkFlag`

`kRTPRssmQueueAndUseMarkerBitFlag`

`kRTPRssmTrackLostPacketsFlag`

`kRTPRssmNoReorderingRequiredFlag`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Your packet reassembler can call this function at any time.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmGetChunkAndIncrRefCount

Causes the base reassembler to create a chunk for you manually.

```
ComponentResult RTPRssmGetChunkAndIncrRefCount (
   RTPReassembler rtpr,
   UInt32 inChunkDataSize,
   const TimeValue64 *inChunkPresentationTime,
   SHChunkRecord **outChunk
);
```

Parameters

rtpr

The component instance of the base reassembler component.

inChunkDataSize

An unsigned 32-bit integer containing the size of the chunk's data portion, in bytes.

inChunkPresentationTime

A pointer to a 64-bit time value specifying the time at which this chunk should be presented, in units of the stream's time scale.

outChunk

On return, contains a pointer to a newly-created `SHChunkRecord` structure.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function is useful if you are overriding the `RTPRssmSendPacketList` (page 144) behavior and constructing the chunk yourself. You must explicitly dispose of the chunk when you are done with it by calling either `RTPRssmDecrChunkRefCount` (page 130) or `RTPRssmSendChunkAndDecrRefCount` (page 143).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPRssmGetExtChunkAndIncrRefCount

Undocumented

```
ComponentResult RTPRssmGetExtChunkAndIncrRefCount (
   RTPReassembler rtpr,
   UInt32 inChunkDataSize,
   const TimeValue64 *inChunkPresentationTime,
   SInt32 inFlags,
   SHExtendedChunkRecord **outChunk
);
```

Parameters

rtpr

Undocumented

inChunkDataSize

Undocumented

inChunkPresentationTime
Undocumented

inFlags
Undocumented

outChunk
A pointer to a pointer to a `SHExtendedChunkRecord` data structure.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6. Can be used only with Mac OS X 10.1 and later.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmGetInfo

Obtains information about your packet reassembler.

```
ComponentResult RTPRssmGetInfo (
   RTPReassembler rtpr,
   OSType inSelector,
   void *ioParams
);
```

Parameters
rtpr
The component instance of your packet reassembler.

inSelector
A selector (see below) for the information desired. See these constants:
`kQTSSourceTrackIDInfo`
`kQTSSourceLayerInfo`
`kQTSSourceLanguageInfo`
`kQTSSourceTrackFlagsInfo`
`kQTSSourceDimensionsInfo`
`kQTSSourceVolumesInfo`
`kQTSSourceMatrixInfo`
`kQTSSourceClipRectInfo`
`kQTSSourceGraphicsModeInfo`
`kQTSSourceScaleInfo`
`kQTSSourceBoundingRectInfo`
`kQTSSourceUserDataInfo`
`kQTSSourceInputMapInfo`

ioParams

A pointer to a data structure appropriate for the type of data requested (see below) . If your component understands the selector, write the requested information into the data structure this parameter points to.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Implement this function only for the selectors you understand. Delegate this function to the base reassembler for any other selectors. The base reassembler will correctly return an error if it doesn't understand the selector either.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPRssmGetPayloadHeaderLength

Obtains the current value of the fixed payload header length from the base reassembler.

```
ComponentResult RTPRssmGetPayloadHeaderLength (
   RTPReassembler rtpr,
   UInt32 *outPayloadHeaderLength
);
```

Parameters

rtpr

The component instance of the base reassembler component.

outPayloadHeaderLength

On return, contains a pointer to an unsigned 32-bit integer containing the length of the payload header in bytes. If your packet reassembler does not implement `RTPRssmAdjustPacketParams` (page 127), or takes no action, the default `payloadHeaderLength` is the fixed header length that is set (default is 0).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Your packet reassembler can call this function at any time.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmGetStreamHandler

Obtains the component instance of the stream handler to which the base reassembler is sending your output.

```
ComponentResult RTPRssmGetStreamHandler (
   RTPReassembler rtpr,
   ComponentInstance *outStreamHandler
);
```

Parameters

rtpr

The component instance of the base reassembler.

outStreamHandler

On return, contains a pointer to the component instance of the stream handler your output is being sent to by the base reassembler.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmGetTimeScale

Obtains the current time scale from the base reassembler.

```
ComponentResult RTPRssmGetTimeScale (
   RTPReassembler rtpr,
   TimeScale *outSHTimeScale
);
```

Parameters

rtpr

The component instance of the base reassembler.

outSHTimeScale

On return, contains a pointer to the time scale in use by the stream handler that is processing your output.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmGetTimeScaleFromPacket

Lets your packet reassembler extract the time scale from a received packet and return it to the base reassembler.

```
ComponentResult RTPRssmGetTimeScaleFromPacket (
   RTPReassembler rtpr,
   QTSStreamBuffer *inStreamBuffer,
   TimeScale *outTimeScale
);
```

Parameters

rtpr
The component instance of your packet reassembler.

inStreamBuffer
A pointer to a received packet from which you may be able to extract a time scale.

outTimeScale
Return a pointer to a valid time scale or return an error. If you return a time scale, the packet will be processed normally. If you return an error, the packet will be discarded.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
If your packet reassembler has not specified a time scale as part of `RTPRssmNewStreamHandler` (page 140), or by calling `RTPRssmSetTimeScale` (page 150), the base reassembler calls this function when it receives packets, which allows your packet reassembler to extract the time scale from a received packet and return it to the base reassembler. Your packet reassembler must set a time scale for the stream handler before the base reassembler can process any incoming packets. If your packet reassembler doesn't know the time scale of its media in advance, because the time scale is contained in the packet header for example, the base reassembler will prompt you for a time scale whenever it receives a packet. If your packet reassembler always uses the same time scale, it should set the time scale when it opens a stream handler, and it does not need to implement this function. The base reassembler will discard received packets until it has been given a valid time scale.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmHandleNewPacket

Called whenever a new packet arrives, giving your packet reassembler the opportunity to process the packet.

```
ComponentResult RTPRssmHandleNewPacket (
   RTPReassembler rtpr,
   QTSStreamBuffer *inStreamBuffer,
   SInt32 inNumWraparounds
);
```

Parameters

rtpr

The component instance of your packet reassembler.

inStreamBuffer

A pointer to the newly-arrived packet.

inNumWraparounds

The upper 32 bits of the 64-bit timestamp (the lower 32 bits are in the RTP packet timestamp).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

You should implement this function only if you need to process the packet yourself, or if you need to extract information from the packets as they arrive (you need to monitor the payload header, for example). If you implement this function, you can process the packet as needed, then delegate the default processing to the base reassembler.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPRssmHasCharacteristic

Determines what features your reassembler supports.

```
ComponentResult RTPRssmHasCharacteristic (
   RTPReassembler rtpr,
   OSType inCharacteristic,
   Boolean *outHasIt
);
```

Parameters

rtpr

The component instance of your packet reassembler.

inCharacteristic

A constant that defines the characteristic being tested.

outHasIt

A pointer to a Boolean value that is TRUE if your packet reassembler has the characteristic, FALSE otherwise.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPRssmIncrChunkRefCount

Tells the base reassembler to keep a copy of the most recent chunk after it has been sent.

```
ComponentResult RTPRssmIncrChunkRefCount (
   RTPReassembler rtpr,
   SHChunkRecord *inChunk
);
```

Parameters

rtpr
The component instance of the base reassembler.

inChunk
A pointer to the chunk record you want to preserve.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function is used to assist in loss recovery, for example. You must call `RTPRssmDecrChunkRefCount` (page 130) to release the chunk when you no longer need it.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPRssmInitialize

Called when the base reassembler is ready to have your packet reassembler begin handling media packets.

```
ComponentResult RTPRssmInitialize (
   RTPReassembler rtpr,
   RTPRssmInitParams *inInitParams
);
```

Parameters

rtpr

The component instance of your packet reassembler

inInitParams

A pointer to an `RTPRssmInitParams` structure. Use the information contained in this structure to initialize your component.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function is not called when the base reassembler opens your component for payload registration information.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTSPketizerReassem

QTSPketizerReassem.win

qtstreaming

qtstreaming.win

Declared In

`QTStreamingComponents.h`

RTPRssmNewStreamHandler

Opens a new stream handler and closes any currently-open stream handler.

```
ComponentResult RTPRssmNewStreamHandler (
   RTPReassembler rtpr,
   OSType inSHType,
   SampleDescriptionHandle inSampleDescription,
   TimeScale inSHTimeScale,
   ComponentInstance *outHandler
);
```

Parameters

rtpr

The component instance of the base reassembler.

inSHType

The stream handler type.

inSampleDescription

A handle to a `SampleDescription` structure appropriate for this media type. Pass in `NIL` if you don't know the `media` type yet. This structure is passed by reference; the caller is responsible for maintaining it.

inSHTimeScale

The time scale for the stream handler to use. Pass in 0 if the time scale is not yet known.

outHandler

On return, contains a pointer to the component instance of the stream handler that has been opened.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

You must pass in a valid `SampleDescription` structure and time scale before the stream handler can process packets. If you do not pass them as part of this function, do so using `RTPRssmSetTimeScale` (page 150) and `RTPRssmSetSampleDescription` (page 148).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTSPketizerReassem

QTSPketizerReassem.win

qtstreaming

qtstreaming.win

Declared In

`QTStreamingComponents.h`

RTPRssmReleasePacketList

Releases memory associated with a packet list that your packet reassembler created itself, or a list your reassembler took ownership of as a result of implementing RTPRssmSendPacketList.

```
ComponentResult RTPRssmReleasePacketList (
   RTPReassembler rtpr,
   RTPRssmPacket *inPacketListHead
);
```

Parameters

rtpr

The component instance of the base reassembler.

inPacketListHead

A pointer to the packet list to dispose of.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This is a housekeeping function that you do not need to perform for packet lists created and handled by the base reassembler, only for packet lists that you create or take ownership of yourself.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmReset

Called to reset all packet reassembler and base reassembler variables for a new run of data.

```
ComponentResult RTPRssmReset (
   RTPReassembler rtpr,
   SInt32 inFlags
);
```

Parameters

rtpr
The component instance of your reassembler.

inFlags
A signed 32-bit integer containing any flags being passed. No flags are currently defined.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function differs from `RTPRssmClearCachedPackets` (page 128), which disposes of the packets but still retains the last sequence number and related information; this function resets all variables as if the reassembler were just opened.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPRssmSendChunkAndDecrRefCount

Called by the packet reassembler when it has finished constructing a chunk and wants the base reassembler to send it to the stream handler.

```
ComponentResult RTPRssmSendChunkAndDecrRefCount (
   RTPReassembler rtpr,
   SHChunkRecord *inChunk,
   const SHServerEditParameters *inServerEdit
);
```

Parameters

rtpr

The component instance of the base reassembler.

inChunk

A pointer to an `SHChunkRecord` structure.

inServerEdit

A pointer to an `SHServerEditParameters` structure containing the server edit parameters. Pass in `NIL` if there is no server edit.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

Use this function to manually send a chunk if you have overridden the default behavior of `RTPRssmSendPacketList` (page 144). This function will decrement the reference count of the chunk.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

RTPRssmSendLostChunk

Allows the base reassembler to send loss notification to the stream handler.

```
ComponentResult RTPRssmSendLostChunk (
   RTPReassembler rtpr,
   const TimeValue64 *inChunkPresentationTime
);
```

Parameters

rtpr

The component instance of the base reassembler.

inChunkPresentationTime

A pointer to a 64-bit time value indicating when the chunk would have been presented, in units of the stream's time scale.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Loss notification is normally performed automatically by the base reassembler. Use this function if you are handling losses or sending chunks manually.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmSendPacketList

Called when the base reassembler is ready to send a sample or chunk based on a list of packets.

```
ComponentResult RTPRssmSendPacketList (
   RTPReassembler rtpr,
   RTPRssmPacket *inPacketListHead,
   const TimeValue64 *inLastChunkPresentationTime,
   SInt32 inFlags
);
```

Parameters

rtpr
The component instance of your packet reassembler.

inPacketListHead
A pointer to the packet list.

inLastChunkPresentationTime
A pointer to a time value which specifies when to present this chunk, in units of the stream's time scale.

inFlags
A signed 32-bit integer containing any flags being passed (see below). See these constants:
`kRTPRssmLostSomePackets`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Implement this call if your packet reassembler needs to modify the packet list, or if it overrides the default handling of packet loss. If you do not implement this call, the base reassembler will adjust the `packet` parameters on all packets in the list, compute the chunk size, and send the chunk. If packet loss has occurred, all the packets will be discarded and the stream handler will be informed that the chunk has been lost.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem

QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPRssmSendStreamBufferRange

Notifies the base reassembler to construct and send a chunk based on a part of the stream buffer.

```
ComponentResult RTPRssmSendStreamBufferRange (
   RTPReassembler rtpr,
   RTPSendStreamBufferRangeParams *inParams
);
```

Parameters

rtpr

The component instance of the base reassembler.

inParams

A pointer to an `RTPSendStreamBufferRangeParams` structure, which specifies the stream buffer, presentation time, start position in the buffer, length of the data in bytes, and any flags.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The contents of the stream buffer will be referenced, not copied. You are responsible for maintaining valid data in the stream buffer.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmSendStreamHandlerChanged

Called when you have changed something in the stream handler and you want the notification propagated.

```
ComponentResult RTPRssmSendStreamHandlerChanged (
   RTPReassembler rtpr
);
```

Parameters

rtpr

The component instance of the base reassembler.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function is useful, for example, if you have changed the dimensions of the video.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmSetCapabilities

Sets the capabilities of a streaming packet reassembler.

```
ComponentResult RTPRssmSetCapabilities (
   RTPReassembler rtpr,
   SInt32 inFlags,
   SInt32 inFlagsMask
);
```

Parameters

rtpr

The component instance of the base reassembler.

inFlags

A signed 32-bit integer containing the logical OR of all the flags (see below) you are setting. See these constants:

`kRTPRssmEveryPacketAChunkFlag`
`kRTPRssmQueueAndUseMarkerBitFlag`
`kRTPRssmTrackLostPacketsFlag`
`kRTPRssmNoReorderingRequiredFlag`

inFlagsMask

Use this field to preserve the state of any flags you do not wish to alter. If a flag (see below) is set in this field, and is not set in the `inFlags` field, it will not be changed from its current setting.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
Your packet reassembler can call this function at any time.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming

qtstreaming.win

Declared In
QTStreamingComponents.h

RTPRssmSetInfo

Sets various parameters of your packet reassembler; it is also called to set parameters of the base reassembler.

```
ComponentResult RTPRssmSetInfo (
   RTPReassembler rtpr,
   OSType inSelector,
   void *ioParams
);
```

Parameters

rtpr

The component instance of your packet reassembler.

inSelector

A selector (see below) for the information being set. Ignore any selectors you do not understand. See these constants:

- kQTSSourceTrackIDInfo
- kQTSSourceLayerInfo
- kQTSSourceLanguageInfo
- kQTSSourceTrackFlagsInfo
- kQTSSourceDimensionsInfo
- kQTSSourceVolumesInfo
- kQTSSourceMatrixInfo
- kQTSSourceClipRectInfo
- kQTSSourceGraphicsModeInfo
- kQTSSourceScaleInfo
- kQTSSourceBoundingRectInfo
- kQTSSourceUserDataInfo
- kQTSSourceInputMapInfo

ioParams

A pointer to the information that should be set.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
Delegate this function to the base reassembler for any selectors you don't understand. If the base reassembler doesn't understand them either, it will return an error to the caller.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPRssmSetPayloadHeaderLength

Called by the packet reassembler to set a fixed header length for your payload.

```
ComponentResult RTPRssmSetPayloadHeaderLength (
   RTPReassembler rtpr,
   UInt32 inPayloadHeaderLength
);
```

Parameters

rtpr

The component instance of the base reassembler.

inPayloadHeaderLength

An unsigned 32-bit integer containing the fixed payload header length, in bytes. If your packet reassembler does not implement RTPRssmAdjustPacketParams (page 127), or takes no action, the default payloadHeaderLength is the fixed header length that is set (default is 0).

Return Value
See Error Codes. Returns noErr if there is no error.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPRssmSetSampleDescription

Changes the SampleDescription structure being used by the stream handler; all subsequent samples will be marked with this new structure.

```
ComponentResult RTPRssmSetSampleDescription (
   RTPReassembler rtpr,
   SampleDescriptionHandle inSampleDescription
);
```

Parameters

rtpr

The component instance of the base reassembler.

inSampleDescription

The handle of a SampleDescription structure to use. You are responsible for keeping the handle and the data structure valid during subsequent operations.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
The `SampleDescription` structure is not passed on a per-packet basis, but a per-sample basis, so the `SampleDescription` structure should not be changed until a complete sample (sometimes called a "frame" or "chunk") has been reassembled.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
QTSPketizerReassem
QTSPketizerReassem.win
qtstreaming
qtstreaming.win

Declared In
`QTStreamingComponents.h`

RTPRssmSetStreamHandler

Assigns a stream handler to the output of the base reassembler.

```
ComponentResult RTPRssmSetStreamHandler (
   RTPReassembler rtpr,
   ComponentInstance inStreamHandler
);
```

Parameters

`rtpr`
The component instance of the base reassembler.

`inStreamHandler`
The component instance of the stream handler to use.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The stream handler must already be opened and initialized, and its time scale must already be set.

Special Considerations

Use this function only if you have opened and initialized a stream handler yourself. The base reassembler will not close the stream handler it is already using.

Version Notes
Introduced in QuickTime 4.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPRssmSetTimeScale

Sets the time scale for the stream handler that will render your output.

```
ComponentResult RTPRssmSetTimeScale (
   RTPReassembler rtpr,
   TimeScale inSHTimeScale
);
```

Parameters

rtpr

The component instance of the base reassembler

inSHTimeScale

The time scale for the stream handler to use

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

The time scale is the number of time units that pass in one second for the media whose sample data is carried in this stream. The stream handler's time scale must be set before it can deliver any data to the user.

Special Considerations

This function is normally used by a packet reassembler when the time scale to use is not initially known. You don't need to call this function if you specified a time scale when the stream handler was opened.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QTStreamingComponents.h`

TerminateQTS

Terminates the QuickTime Streaming toolbox.

```
OSErr TerminateQTS (
   void
);
```

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

Callbacks

QTSNotificationProc

A back channel from a presentation to its creator, sending notification of various events such as a presentation, ending, or acknowledgment of a preroll request.

```
typedef ComponentResult (*QTSNotificationProcPtr) (ComponentResult inErr, OSType
inNotificationType, void *inNotificationParams, void *inRefCon);
```

If you name your function `MyQTSNotificationProc`, you would declare it this way:

```
ComponentResult MyQTSNotificationProc (
    ComponentResult    inErr,
    OSType             inNotificationType,
    void               *inNotificationParams,
    void               *inRefCon );
```

Parameters

inErr

Undocumented

inNotificationType

The kind of notification; see `QuickTimeStreaming.h`.

inNotificationParams

Undocumented

inRefCon

Undocumented

Return Value

See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPMPDataReleaseProc

Routine called when a media packetizer is finished with its sample data.

```
typedef void (*RTPMPDataReleaseProcPtr) (UInt8 *inData, void *inRefCon);
```

If you name your function `MyRTPMPDataReleaseProc`, you would declare it this way:

```
void MyRTPMPDataReleaseProc (
    UInt8    *inData,
    void     *inRefCon );
```

Parameters

inData

A pointer to the data.

inRefCon
A pointer to information passed from a `RTPMPSampleDataParams` structure.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPPBCallbackProc

Routine used to communicate with the caller of a media packetizer.

```
typedef void (*RTPPBCallbackProcPtr) (OSType inSelector, void *ioParams, void
*inRefCon);
```

If you name your function `MyRTPPBCallbackProc`, you would declare it this way:

```
void MyRTPPBCallbackProc (
    OSType    inSelector,
    void      *ioParams,
    void      *inRefCon );
```

Parameters

inSelector
Undocumented

ioParams
Undocumented

inRefCon
Undocumented

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

Data Types

MediaPacketizerRequirements

Stores the functional requirements for a media packetizer.

```
struct MediaPacketizerRequirements {
     OSType    mediaType;
     OSType    dataFormat;
     UInt32    capabilityFlags;
     UInt8     canPackMatrixType;
     UInt8     pad[3];
 };
```

Fields
`mediaType`

Discussion
Media type required; see `Data References`. 0 means all media types.

`dataFormat`

Discussion
Data format required; see `Media Identifiers`. 0 means all formats.

`capabilityFlags`

Discussion
Constants (see below) that indicate the packetizer's ability to handle non-standard track characteristics. See these constants:

- `kMediaPacketizerCanPackEditRate`
- `kMediaPacketizerCanPackLayer`
- `kMediaPacketizerCanPackVolume`
- `kMediaPacketizerCanPackBalance`
- `kMediaPacketizerCanPackGraphicsMode`
- `kMediaPacketizerCanPackEmptyEdit`

`canPackMatrixType`

Discussion
Constant (see below); the packetizer needs to pack any matrix type up to this level. Set to `identityMatrixType` for identity matrix (no translation) only. See these constants:

- `identityMatrixType`
- `translateMatrixType`
- `scaleMatrixType`
- `scaleTranslateMatrixType`
- `linearMatrixType`
- `linearTranslateMatrixType`
- `perspectiveMatrixType`

`pad`

Discussion
Unused.

Related Functions
`QTSFindMediaPacketizer` (page 20)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

MediaPacketizerRequirementsPtr

Represents a type used by the QuickTime Streaming API.

```
typedef MediaPacketizerRequirements * MediaPacketizerRequirementsPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

QTAtomSpec

Specifies an atom and its container.

```
struct QTAtomSpec {
     QTAtomContainer    container;
     QTAtom             atom;
 };
```

Fields
`container`

Discussion
A QT atom container.

`atom`

Discussion
A QT atom.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTAtomSpecPtr

Represents a type used by the QuickTime Streaming API.

```
typedef QTAtomSpec * QTAtomSpecPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

QTSExportParams

Undocumented

```
struct QTSExportParams {
     SInt32               version;
     OSType               exportType;
     void                 *exportExtraData;
     OSType               destinationContainerType;
     void                 *destinationContainerData;
     void                 *destinationContainerExtras;
     SInt32               flagsIn;
     SInt32               flagsOut;
     QTSModalFilterUPP    filterProc;
     void                 *filterProcRefCon;
     Component            exportComponent;
 };
```

Fields
`version`

Discussion
Undocumented

`exportType`

Discussion
Undocumented

`exportExtraData`

Discussion
Undocumented

`destinationContainerType`

Discussion
Undocumented

`destinationContainerData`

Discussion
Undocumented

`destinationContainerExtras`

Discussion
Undocumented

`flagsIn`

Discussion
Undocumented

`flagsOut`

Discussion
Undocumented

`filterProc`

Discussion
Undocumented

`filterProcRefCon`

Discussion
Undocumented

`exportComponent`

Discussion
Undocumented

Related Functions
`QTSPresExport` (page 49)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSInstantOnPref

Contains instant on information for QuickTime Streaming.

```
struct QTSInstantOnPref {
     SInt32    flags;
     SInt32    factor;
 };
```

Fields
`flags`

Discussion
Constants (see below) that enable instant on. See these constants:
`kQTSInstantOnFlag_Enable`
`kQTSInstantOnFlag_Permitted`

`factor`

Discussion
Values can range from 0 to 100; the default value is 50.

Version Notes
Introduced in QuickTime 6.

Related Functions
`QTSPrefsGetInstantOnSettings` (page 47)
`QTSPrefsSetInstantOnSettings` (page 47)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSMediaParams

Combines the QTSVideoParams and QTSAudioParams structures.

```
struct QTSMediaParams {
     QTSVideoParams    v;
     QTSAudioParams    a;
 };
```

Fields
`v`

Discussion
A `QTSVideoParams` structure.

`a`

Discussion
A `QTSAudioParams` structure.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSMemPtr

Represents a type used by the QuickTime Streaming API.

```
typedef struct OpaqueQTSMemPtr * QTSMemPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSNewPresentationParams

Specifies a presentation for QTSNewPresentation.

```
struct QTSNewPresentationParams {
     OSType                dataType;
     const void *          data;
     UInt32                dataLength;
     QTSEditListHandle     editList;
     SInt32                flags;
     TimeScale             timeScale;
     QTSMediaParams *      mediaParams;
     QTSNotificationUPP    notificationProc;
     void *                notificationRefCon;
 };
```

Fields
`dataType`

Discussion
Undocumented

`data`

Discussion
Undocumented

`dataLength`

Discussion
Undocumented

`editList`

Discussion
A handle to a `QTSEditList` structure.

`flags`

Discussion
Undocumented

`timeScale`

Discussion
The time scale; set to 0 for the default time scale.

`mediaParams`

Discussion
Undocumented

`notificationProc`

Discussion
A pointer to a `QTSNotificationProc` callback.

`notificationRefCon`

Discussion
A reference constant to be passed to the `QTSNotificationProc` callback.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSNoProxyPref

Provides data for the QTSPrefsGetNoProxyURLs function.

```
struct QTSNoProxyPref {
     UInt32    flags;
     UInt32    seed;
     char      urlList[1];
 };
```

Fields
`flags`

Discussion
Undocumented

`seed`

Discussion
A seed value from the last time this setting was read from the system preferences.

`urlList`

Discussion
A null-terminated, comma-delimited list of URLs.

Related Functions
`QTSPrefsGetNoProxyURLs` (page 47)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSNotificationUPP

Represents a type used by the QuickTime Streaming API.

```
typedef STACK_UPP_TYPE(QTSNotificationProcPtr) QTSNotificationUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresentation

Represents a type used by the QuickTime Streaming API.

```
typedef QTSPresentationRecord * QTSPresentation;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSPresentationRecord

Defines a presentation.

```
struct QTSPresentationRecord {
     long    data[1];
 };
```

Fields
`data`

Discussion
Array of data that constitutes the presentation.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSPresIdleParams

Provides parameters for QTSPresIdle.

```
struct QTSPresIdleParams {
     QTSStream      stream;
     TimeValue64    movieTimeToDisplay;
     SInt32         flagsIn;
     SInt32         flagsOut;
 };
```

Fields
`stream`

Discussion
A pointer to a `QTSStreamRecord` structure.

`movieTimeToDisplay`

Discussion
Undocumented

`flagsIn`

Discussion
Undocumented

`flagsOut`

Discussion
Undocumented

Related Functions
`QTSPresIdle` (page 66)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSPresParams

Undocumented.

```
struct QTSPresParams {
     UInt32                version;
     QTSEditListHandle     editList;
     SInt32                flags;
     TimeScale             timeScale;
     QTSMediaParams        *mediaParams;
     QTSNotificationUPP    notificationProc;
     void                  *notificationRefCon;
 };
```

Fields
`version`

Discussion
Undocumented

`editList`

Discussion
Undocumented

`flags`

Discussion
Undocumented

`timeScale`

Discussion
Undocumented

`mediaParams`

Discussion
Undocumented

`notificationProc`

Discussion
Undocumented

`notificationRefCon`

Discussion
Undocumented

Related Functions
QTSNewPresentationFromData (page 36)
QTSNewPresentationFromDataRef (page 37)
QTSNewPresentationFromFile (page 37)

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

QTSProxyPref

Provides data for the QTSPrefsFindProxyByType function.

```
struct QTSProxyPref {
     UInt32    flags;
     SInt32    portID;
     UInt32    seed;
     Str255    serverNameStr;
 };
```

Fields
flags

Discussion
Undocumented

portID

Discussion
ID of the port to use for this connection type.

seed

Discussion
A seed value from the last time this setting was read from the system preferences.

serverNameStr

Discussion
A proxy server URL.

Related Functions
QTSPrefsFindProxyByType (page 44)

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

QTSSourcer

Represents a type used by the QuickTime Streaming API.

```
typedef ComponentInstance QTSSourcer;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

QTSSourcerInitParams

Holds information for initializing a streaming sourcer.

```
struct QTSSourcerInitParams {
     SInt32    version;
     SInt32    flags;
     OSType    dataType;
     void      *data;
     UInt32    dataLength;
 };
```

Fields

`version`

Discussion
Undocumented

`flags`

Discussion
Undocumented

`dataType`

Discussion
Undocumented

`data`

Discussion
Undocumented

`dataLength`

Discussion
Undocumented

Related Functions

`QTSNewSourcer` (page 39)
`QTSSourcerInitialize` (page 87)

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSStatHelper

Represents a type used by the QuickTime Streaming API.

```
typedef QTSStatHelperRecord * QTSStatHelper;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeStreaming.h`

QTSStatHelperNextParams

Holds information about the next streaming statistic obtained by QTSStatHelperNext.

```
struct QTSStatHelperNextParams {
     SInt32       flags;
     OSType       returnedStatisticsType;
     QTSStream    returnedStream;
     UInt32       maxStatNameLength;
     char *       returnedStatName;
     UInt32       maxStatStringLength;
     char *       returnedStatString;
     UInt32       maxStatUnitLength;
     char *       returnedStatUnit;
 };
```

Fields

`flags`

Discussion
Undocumented See these constants:
`kQTSStatHelperReturnPascalStringsFlag`

`returnedStatisticsType`

Discussion
Undocumented

`returnedStream`

Discussion
On return, a pointer to a `QTSStreamRecord` structure.

`maxStatNameLength`

Discussion
Undocumented

`returnedStatName`

Discussion
Undocumented; pass `NIL` if you don't want this information.

`maxStatStringLength`

Discussion
Undocumented

`returnedStatString`

Discussion
Undocumented; pass `NIL` if you don't want this information.

`maxStatUnitLength`

Discussion
Undocumented

`returnedStatUnit`

Discussion
Undocumented; pass `NIL` if you don't want this information.

Discussion
When you call `QTSStatHelperNext` (page 91), specifying a statistic helper and the address of this structure, QuickTime fills in this structure with information about the next statistic obtained by the statistic helper.

Related Functions
`QTSStatHelperNext` (page 91)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSStatHelperRecord

Defines the component instance of a statistics helper.

```
struct QTSStatHelperRecord {
     long    data[1];
 };
```

Fields
`data`

Discussion
The component instance of the statistics helper.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSStream

Represents a type used by the QuickTime Streaming API.

```
typedef QTSStreamRecord * QTSStream;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeStreaming.h`

QTSStreamBuffer

Defines a stream buffer for QuickTime streaming.

```
struct QTSStreamBuffer {
     struct QTSStreamBuffer *   reserved1;
     struct QTSStreamBuffer *   reserved2;
     struct QTSStreamBuffer *   next;
     unsigned char *            rptr;
     unsigned char *            wptr;
     long                       reserved3;
     UInt32                     metadata[4];
     SInt32                     flags;
 };
```

Fields

`reserved1`

Discussion
Reserved; do not use.

`reserved2`

Discussion
Reserved; do not use.

`next`

Discussion
A pointer to the next message block in a message.

`rptr`

Discussion
A pointer to the first byte in the data buffer that contains real data.

`wptr`

Discussion
A pointer to the byte after the last byte in the data buffer that contains real data.

`reserved3`

Discussion
Reserved; do not use.

`metadata`

Discussion
Usage defined by message sender.

`flags`

Discussion
Reserved; do not use.

Related Functions

`QTSCopyMessage` (page 17)
`QTSDupMessage` (page 20)
`QTSFlattenMessage` (page 24)
`QTSFreeMessage` (page 25)
`QTSMessageLength` (page 34)
`RTPRssmGetTimeScaleFromPacket` (page 137)
`RTPRssmHandleNewPacket` (page 138)

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSStreamRecord

Contains a stream for QuickTime streaming.

```
struct QTSStreamRecord {
     long    data[1];
 };
```

Fields
`data`

Discussion
An array of data representing the stream.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSTransportPref

Records streaming transport preferences.

```
struct QTSTransportPref {
     OSType    protocol;
     SInt32    portID;
     UInt32    flags;
     UInt32    seed;
 };
```

Fields
`protocol`

Discussion
Constant that identifies the streaming transport protocol; see `Streaming Transport Atoms`.

`portID`

Discussion
ID of the port to use for this connection type.

`flags`

Discussion
Connection flags (see below). See these constants:

`kConnectionActive`

`kConnectionUseSystemPref`

`seed`

Discussion
A seed value from the last time this setting was read from the system preferences.

Related Functions
`QTSPrefsFindConnectionByType` (page 43)
`QTSPrefsGetActiveConnection` (page 46)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPMediaPacketizer

Represents a type used by the QuickTime Streaming API.

```
typedef ComponentInstance RTPMediaPacketizer;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPDataReleaseUPP

Represents a type used by the QuickTime Streaming API.

```
typedef STACK_UPP_TYPE(RTPMPDataReleaseProcPtr) RTPMPDataReleaseUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPMPSampleDataParams

Holds media packetizer sample data, including any number of samples or a partial sample.

```
struct RTPMPSampleDataParams {
     UInt32                 version;
     UInt32                 timeStamp;
     UInt32                 duration;
     UInt32                 playOffset;
     Fixed                  playRate;
     SInt32                 flags;
     UInt32                 sampleDescSeed;
     Handle                 sampleDescription;
     RTPMPSampleRef         sampleRef;
     UInt32                 dataLength;
     const UInt8 *          data;
     RTPMPDataReleaseUPP    releaseProc;
     void *                 refCon;
 };
```

Fields
`version`

Discussion
Version of the data structure. Currently always 0.

`timeStamp`

Discussion
RTP time stamp for the presentation of the sample data. This time stamp has already been adjusted by edits, edit rates, etc.

`duration`

Discussion
`Duration` (in RTP time scale) of the sample. For unknown duration, enter 0.

`playOffset`

Discussion
Offset within the media sample itself. This is only used for media formats where a single media sample can span across multiple time units. QuickTime Music is an example of this, where a single sample spans the entire track. For most video and audio formats, this will be 0.

`playRate`

Discussion
1.0 (0x00010000) is normal. Higher numbers indicate faster play rates. Note that timeStamp is already adjusted by the rate. This field is generally of interest only to audio packetizers.

`flags`

Discussion
Flag (see below) to indicate if the sample is a sync sample (key frame). See these constants:
`kRTPMPSyncSampleFlag`

`sampleDescSeed`

Discussion
If the sample description changes, this number will change.

`sampleDescription`

Discussion
The sample description for the given media sample.

`sampleRef`

Discussion
Reserved; do not use.

`dataLength`

Discussion
`Size` of the media data.

`data`

Discussion
Pointer to the media data.

`releaseProc`

Discussion
If not `NIL`, you need to call your `RTPMPDataReleaseProc` when you are finished with the sample data.

`refCon`

Discussion
Information to pass to the `RTPMPDataReleaseProc`.

Related Functions
`RTPPBAddPacketSampleData` (page 113)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPPacketBuilder

Represents a type used by the QuickTime Streaming API.

```
typedef ComponentInstance RTPPacketBuilder;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPPacketGroupRef

Represents a type used by the QuickTime Streaming API.

```
typedef struct OpaqueRTPPacketGroupRef * RTPPacketGroupRef;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPPacketRef

Represents a type used by the QuickTime Streaming API.

```
typedef struct OpaqueRTPPacketRef * RTPPacketRef;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPPacketRepeatedDataRef

Represents a type used by the QuickTime Streaming API.

```
typedef struct OpaqueRTPPacketRepeatedDataRef * RTPPacketRepeatedDataRef;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPPayloadSortRequest

Specifies the sort order for a list of packetizers.

```
struct RTPPayloadSortRequest {
     long                        characteristicCount;
     RTPPayloadCharacteristic    characteristic[1];
 };
```

Fields
`characteristicCount`

Discussion
The number of structures in the characteristic field.

`characteristic`

Discussion
An array of `RTPPayloadCharacteristic` structures.

Related Functions
`QTSFindMediaPacketizer` (page 20)
`QTSFindMediaPacketizerForPayloadID` (page 21)
`QTSFindMediaPacketizerForPayloadName` (page 21)
`QTSFindMediaPacketizerForTrack` (page 22)
`QTSFindReassemblerForPayloadID` (page 23)
`QTSFindReassemblerForPayloadName` (page 23)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPPayloadSortRequestPtr

Represents a type used by the QuickTime Streaming API.

```
typedef RTPPayloadSortRequest * RTPPayloadSortRequestPtr;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPPBCallbackUPP

Represents a type used by the QuickTime Streaming API.

```
typedef STACK_UPP_TYPE(RTPPBCallbackProcPtr) RTPPBCallbackUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QTStreamingComponents.h`

RTPReassembler

Represents a type used by the QuickTime Streaming API.

```
typedef ComponentInstance RTPReassembler;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QTStreamingComponents.h

RTPRssmInitParams

Initializes a packet reassembler component.

```
struct RTPRssmInitParams {
     RTPSSRC      ssrc;
     UInt8        payloadType;
     UInt8        pad[3];
     TimeBase     timeBase;
     TimeScale    timeScale;
 };
```

Fields
ssrc

Discussion
Undocumented

payloadType

Discussion
Undocumented

pad

Discussion
Unused.

timeBase

Discussion
A reference to the reassembler's time base. You obtain a time base by calling GetMovieTimeBase or NewTimeBase.

timeScale

Discussion
The reassembler's time scale.

Related Functions
RTPRssmInitialize (page 140)

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

RTPRssmPacket

A streaming reassembler packet list.

```
struct RTPRssmPacket {
     struct RTPRssmPacket *    next;
     struct RTPRssmPacket *    prev;
     QTSStreamBuffer *         streamBuffer;
     Boolean                   paramsFilledIn;
     UInt8                     pad[1];
     UInt16                    sequenceNum;
     UInt32                    transportHeaderLength;
     UInt32                    payloadHeaderLength;
     UInt32                    dataLength;
     SHServerEditParameters    serverEditParams;
     TimeValue64               timeStamp;
     SInt32                    chunkFlags;
     SInt32                    flags;
 };
```

Fields

`next`

Discussion
A pointer to the next `RTPRssmPacket` structure.

`prev`

Discussion
A pointer to the previous `RTPRssmPacket` structure.

`streamBuffer`

Discussion
A pointer to a `QTSStreamBuffer` structure defining the stream buffer.

`paramsFilledIn`

Discussion
Undocumented

`pad`

Discussion
Undocumented

`sequenceNum`

Discussion
Undocumented

`transportHeaderLength`

Discussion
Undocumented

`payloadHeaderLength`

Discussion
Undocumented

`dataLength`

Discussion
Undocumented

`serverEditParams`

Discussion
Undocumented

`timeStamp`

Discussion
Undocumented

`chunkFlags`

Discussion
Undocumented

`flags`

Discussion
Undocumented

Related Functions
`RTPRssmAdjustPacketParams` (page 127)
`RTPRssmComputeChunkSize` (page 129)
`RTPRssmCopyDataToChunk` (page 129)
`RTPRssmFillPacketListParams` (page 131)
`RTPRssmReleasePacketList` (page 141)
`RTPRssmSendPacketList` (page 144)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPSendStreamBufferRangeParams

Undocumented

```
struct RTPSendStreamBufferRangeParams {
     QTSStreamBuffer *                streamBuffer;
     TimeValue64                      presentationTime;
     UInt32                           chunkStartPosition;
     UInt32                           numDataBytes;
     SInt32                           chunkFlags;
     SInt32                           flags;
     const SHServerEditParameters *   serverEditParams;
 };
```

Fields
`streamBuffer`

Discussion
Undocumented

`presentationTime`

Discussion
Undocumented

`chunkStartPosition`

Discussion
Undocumented

`numDataBytes`

Discussion
Undocumented

`chunkFlags`

Discussion
Undocumented

`flags`

Discussion
Undocumented

`serverEditParams`

Discussion
Undocumented

Related Functions
`RTPRssmSendStreamBufferRange` (page 145)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

SHChunkRecord

Defines a chunk for a reassembler.

```
struct SHChunkRecord {
     UInt32                          version;
     long                            reserved1;
     SInt32                          flags;
     UInt32                          dataSize;
     const UInt8 *                   dataPtr;
     long                            reserved2;
     long                            reserved3;
     TimeValue64                     presentationTime;
     long                            reserved4;
     long                            reserved5;
     const SHServerEditParameters *  serverEditParameters;
     long                            reserved6;
     long                            reserved7;
 };
```

Fields
`version`

Discussion
Undocumented

`reserved1`

Discussion
Reserved; do not use.

`flags`

Discussion
Undocumented

`dataSize`

Discussion
The size of the chunk data.

`dataPtr`

Discussion
A pointer to the chunk data.

`reserved2`

Discussion
Reserved; do not use.

`reserved3`

Discussion
Reserved; do not use.

`presentationTime`

Discussion
Undocumented

`reserved4`

Discussion
Reserved; do not use.

`reserved5`

Discussion
Reserved; do not use.

`serverEditParameters`

Discussion
A pointer to an `SHServerEditParameters` structure containing the server edit parameters

`reserved6`

Discussion
Reserved; do not use.

`reserved7`

Discussion
Reserved; do not use.

Related Functions

`RTPRssmCopyDataToChunk` (page 129)
`RTPRssmDecrChunkRefCount` (page 130)
`RTPRssmGetChunkAndIncrRefCount` (page 132)
`RTPRssmIncrChunkRefCount` (page 139)
`RTPRssmSendChunkAndDecrRefCount` (page 143)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

SHExtendedChunkRecord

Extends an SHChunkRecord data structure.

```
struct SHExtendedChunkRecord {
     SHChunkRecord    chunk;
     SInt32           extendedFlags;
     SInt32           extendedData[10];
 };
```

Fields
`chunk`

Discussion
A `SHChunkRecord` data structure.

`extendedFlags`

Discussion
Constants (see below) that indicate what data is being added. See these constants:
`kSHExtendedChunkFlag_HasSampleCount`
`kSHExtendedChunkFlag_HasFrameLengths`

`extendedData`

Discussion
The additional data.

Version Notes
Introduced in QuickTime 6.

Related Functions
`RTPRssmGetExtChunkAndIncrRefCount` (page 133)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

SHServerEditParameters

Undocumented

```
struct SHServerEditParameters {
     UInt32         version;
     Fixed          editRate;
     TimeValue64    dataStartTime_mediaAxis;
     TimeValue64    dataEndTime_mediaAxis;
 };
```

Fields
`version`

Discussion
Undocumented

`editRate`

Discussion
Undocumented

`dataStartTime_mediaAxis`

Discussion
Undocumented

`dataEndTime_mediaAxis`

Discussion
Undocumented

Version Notes
Introduced in QuickTime 6.

Related Functions
`RTPRssmGetExtChunkAndIncrRefCount` (page 133)

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

Constants

MediaPacketizerRequirements Values

Constants passed to MediaPacketizerRequirements.

```
enum {
  identityMatrixType            = 0x00, /* result if matrix is identity */
  translateMatrixType           = 0x01, /* result if matrix translates */
  scaleMatrixType               = 0x02, /* result if matrix scales */
  scaleTranslateMatrixType      = 0x03, /* result if matrix scales and translates
 */
  linearMatrixType              = 0x04, /* result if matrix is general 2 x 2 */
  linearTranslateMatrixType     = 0x05, /* result if matrix is general 2 x 2 and
translates */
  perspectiveMatrixType         = 0x06  /* result if matrix is general 3 x 3 */
};
enum {
  kMediaPacketizerCanPackEditRate = 1 << 0,
  kMediaPacketizerCanPackLayer  = 1 << 1,
  kMediaPacketizerCanPackVolume = 1 << 2,
  kMediaPacketizerCanPackBalance = 1 << 3,
  kMediaPacketizerCanPackGraphicsMode = 1 << 4,
  kMediaPacketizerCanPackEmptyEdit = 1 << 5
};
```

Constants

`identityMatrixType`
Matrix is identity; value is 0x00.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.

`translateMatrixType`
Matrix translates; value is 0x01.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.

`scaleMatrixType`
Matrix scales; value is 0x02.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`scaleTranslateMatrixType`
Matrix translates and scales; value is 0x03.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`linearMatrixType`
Matrix is general 2 x 2 type; value is 0x04.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`linearTranslateMatrixType`
Matrix is general 2 x 2 type and translates; value is 0x05

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`perspectiveMatrixType`
Matrix is general 3 x 3 type; value is 0x06.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kMediaPacketizerCanPackEditRate`
The packetizer can pack the edit rate value.

Available in Mac OS X v10.0 and later.

Declared in `QTStreamingComponents.h`.

`kMediaPacketizerCanPackLayer`
The packetizer can pack the layer number.

Available in Mac OS X v10.0 and later.

Declared in `QTStreamingComponents.h`.

`kMediaPacketizerCanPackVolume`
The packetizer can pack the sound volume value.

Available in Mac OS X v10.0 and later.

Declared in `QTStreamingComponents.h`.

`kMediaPacketizerCanPackBalance`
The packetizer can pack the sound balance value.

Available in Mac OS X v10.0 and later.

Declared in `QTStreamingComponents.h`.

`kMediaPacketizerCanPackGraphicsMode`
The packetizer can pack the graphics transfer mode value.

Available in Mac OS X v10.0 and later.

Declared in `QTStreamingComponents.h`.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSTransportPref Values

Constants passed to QTSTransportPref.

```
enum {
  kConnectionActive             = (1L << 0),
  kConnectionUseSystemPref      = (1L << 1)
};
```

Constants

`kConnectionActive`

The connection is active.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeStreaming.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSStatisticsParams Values

Constants passed to QTSStatisticsParams.

```
enum {
  kQTSAllStatisticsType         = 'all ',
  kQTSShortStatisticsType       = 'shrt',
  kQTSSummaryStatisticsType     = 'summ'
};
```

Constants

`kQTSAllStatisticsType`

A full statistics helper for all statistics; constant value is `'all '`.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeStreaming.h`.

`kQTSShortStatisticsType`

A short statistics helper; constant value is `'shrt'`.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeStreaming.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSPresGetFlags Values

Constants passed to QTSPresGetFlags.

```
enum {
  kQTSAutoModeFlag              = 0x00000001,
  kQTSDontShowStatusFlag        = 0x00000008,
  kQTSSendMediaFlag             = 0x00010000,
  kQTSReceiveMediaFlag          = 0x00020000
};
```

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSPrefsGetActiveConnection Values

Constants passed to QTSPrefsGetActiveConnection.

```
enum {
  kQTSDirectConnectHTTPProtocol = 'http',
  kQTSDirectConnectRTSPProtocol = 'rtsp'
};
```

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

kQTSDontGetDataStatisticsFlag

Constants grouped with kQTSDontGetDataStatisticsFlag.

```
enum {
  kQTSGetNameStatisticsFlag     = 0x00000001,
  kQTSDontGetDataStatisticsFlag = 0x00000002,
  kQTSUpdateAtomsStatisticsFlag = 0x00000004,
  kQTSGetUnitsStatisticsFlag    = 0x00000008,
  kQTSUpdateAllIfNecessaryStatisticsFlag = 0x00010000
};
```

Constants

`kQTSGetUnitsStatisticsFlag`

The statistics helper is to get units statistics.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeStreaming.h`.

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSPresSetInfo Values

Constants passed to QTSPresSetInfo.

```
enum {
  kQTSGetURLLink                = 'gull' /* QTSGetURLLinkRecord* */
};
```

Declared In
`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSInstantOnPref Values

Constants passed to QTSInstantOnPref.

```
enum {
  kQTSInstantOnFlag_Enable     = (1L << 0), /* instant on is enabled (read/write)*/
  kQTSInstantOnFlag_Permitted   = (1L << 1) /* instant on is possible (read only)*/
};
```

Constants

`kQTSInstantOnFlag_Enable`

Instant on is enabled for read or write operations.

Available in Mac OS X v10.2 and later.

Declared in `QuickTimeStreaming.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSMediaSetInfo Values

Constants passed to QTSMediaSetInfo.

```
enum {
  kQTSMediaPresentationInfo     = 'pres', /* QTSMediaPresentationParams* */
  kQTSMediaNotificationInfo     = 'noti', /* QTSMediaNotificationParams* */
  kQTSMediaTotalDataRateInfo    = 'dtrt', /* UInt32*, bits/sec */
  kQTSMediaLostPercentInfo      = 'lspc', /* Fixed* */
  kQTSMediaNumStreamsInfo       = 'nstr', /* UInt32* */
  kQTSMediaIndSampleDescriptionInfo = 'isdc' /* QTSMediaIndSampleDescriptionParams*
 */
};
```

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSNewPtr Values

Constants passed to QTSNewPtr.

```
enum {
  kQTSMemAllocAllocatedInTempMem = 0x00000001,
  kQTSMemAllocAllocatedInSystemMem = 0x00000002
};
enum {
  kQTSMemAllocClearMem          = 0x00000001,
  kQTSMemAllocDontUseTempMem    = 0x00000002,
  kQTSMemAllocTryTempMemFirst   = 0x00000004,
  kQTSMemAllocDontUseSystemMem  = 0x00000008,
  kQTSMemAllocTrySystemMemFirst = 0x00000010,
  kQTSMemAllocHoldMemory        = 0x00001000,
  kQTSMemAllocIsInterruptTime   = 0x01010000 /* currently not supported for alloc*/
};
```

Constants

`kQTSMemAllocAllocatedInSystemMem`

The block was allocated in system memory.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeStreaming.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSSetNetworkAppName Values

Constants passed to QTSSetNetworkAppName.

```
enum {
  kQTSNetworkAppNameIsFullNameFlag = 0x00000001
};
```

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSStatHelperNextParams Values

Constants passed to QTSStatHelperNextParams.

```
enum {
  kQTSStatHelperReturnPascalStringsFlag = 0x00000001
};
```

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

QTSInsertStatisticUnits Values

Constants passed to QTSInsertStatisticUnits.

```
enum {
  kQTSStatisticsNoUnitsType     = 0,
  kQTSStatisticsPercentUnitsType = 'pcnt',
  kQTSStatisticsBitsPerSecUnitsType = 'bps ',
  kQTSStatisticsFramesPerSecUnitsType = 'fps '
};
enum {
  kQTSStatisticsStreamAtomType  = 'strm',
  kQTSStatisticsNameAtomType    = 'name', /* chars only, no length or terminator
*/
  kQTSStatisticsDataFormatAtomType = 'frmt', /* OSType */
  kQTSStatisticsDataAtomType    = 'data',
  kQTSStatisticsUnitsAtomType   = 'unit', /* OSType */
  kQTSStatisticsUnitsNameAtomType = 'unin' /* chars only, no length or terminator
 */
};
```

Constants

`kQTSStatisticsFramesPerSecUnitsType`

Frames-per-second unit type; value is `'fps '`.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeStreaming.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

kQTSStatisticsFixedDataFormat

Constants grouped with kQTSStatisticsFixedDataFormat.

```
enum {
  kQTSStatisticsSInt32DataFormat = 'si32',
  kQTSStatisticsUInt32DataFormat = 'ui32',
  kQTSStatisticsSInt16DataFormat = 'si16',
  kQTSStatisticsUInt16DataFormat = 'ui16',
  kQTSStatisticsFixedDataFormat = 'fixd',
  kQTSStatisticsUnsignedFixedDataFormat = 'ufix',
  kQTSStatisticsStringDataFormat = 'strg',
  kQTSStatisticsOSTypeDataFormat = 'ostp',
  kQTSStatisticsRectDataFormat  = 'rect',
  kQTSStatisticsPointDataFormat = 'pont'
};
```

Constants

`kQTSStatisticsOSTypeDataFormat`

`OSType` (32-bit) format; value is `'ostp'`.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeStreaming.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

Streaming Transport Atoms

Identify transport atom types for QuickTime streaming.

```
enum {
  kQTSTransAndProxyAtomType     = 'strp', /* transport/proxy prefs root atom*/
  kQTSConnectionPrefsVersion    = 'vers', /*   prefs format version*/
  kQTSTransportPrefsAtomType    = 'trns', /*   tranport prefs root atom*/
  kQTSConnectionAtomType        = 'conn', /*     connection prefs atom type, one
for each transport type*/
  kQTSUDPTransportType          = 'udp ', /*     udp transport prefs*/
  kQTSHTTPTransportType         = 'http', /*     http transport prefs*/
  kQTSTCPTransportType          = 'tcp ', /*     tcp transport prefs    */
  kQTSProxyPrefsAtomType        = 'prxy', /*   proxy prefs root atom*/
  kQTSHTTPProxyPrefsType        = 'http', /*     http proxy settings*/
  kQTSRTSPProxyPrefsType        = 'rtsp', /*     rtsp proxy settings*/
  kQTSSOCKSProxyPrefsType       = 'sock', /*     socks proxy settings*/
  kQTSProxyUserInfoPrefsType    = 'user', /*   proxy username/password root atom*/
  kQTSDontProxyPrefsAtomType    = 'nopr', /*   no-proxy prefs root atom*/
  kQTSDontProxyDataType         = 'data', /*     no proxy settings*/
  kQTSInstantOnPrefsAtomType    = 'inon' /* instant on prefs*/
};
```

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

kRTPMPHasUserSettingsDialogCharacteristic

Constants grouped with kRTPMPHasUserSettingsDialogCharacteristic.

```
enum {
  kRTPMPNoSampleDataRequiredCharacteristic = 'nsdr',
  kRTPMPHasUserSettingsDialogCharacteristic = 'sdlg',
  kRTPMPPrefersReliableTransportCharacteristic = 'rely',
  kRTPMPRequiresOutOfBandDimensionsCharacteristic = 'robd',
  kRTPMPReadsPartialSamplesCharacteristic = 'rpsp'
};
```

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

kRTPInfo_FormatString

Constants grouped with kRTPInfo_FormatString.

```
enum {
  kRTPMPPayloadTypeInfo         = 'rtpp', /* RTPMPPayloadTypeParams* */
  kRTPMPRTPTimeScaleInfo        = 'rtpt', /* TimeScale* */
  kRTPMPRequiredSampleDescriptionInfo = 'sdsc', /* SampleDescriptionHandle* */
  kRTPMPMinPayloadSize          = 'mins', /* UInt32* in bytes, does not include
rtp header; default is 0 */
  kRTPMPMinPacketDuration       = 'mind', /* UInt3* in milliseconds; default is no
 min required */
  kRTPMPSuggestedRepeatPktCountInfo = 'srpc', /* UInt32* */
  kRTPMPSuggestedRepeatPktSpacingInfo = 'srps', /* UInt32* in milliseconds */
  kRTPMPMaxPartialSampleSizeInfo = 'mpss', /* UInt32* in bytes */
  kRTPMPPreferredBufferDelayInfo = 'prbd', /* UInt32* in milliseconds */
  kRTPMPPayloadNameInfo         = 'name', /* StringPtr */
  kRTPInfo_FormatString         = 'fmtp' /* char **, caller allocates ptr, callee
 disposes */
};
```

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

RTPMPInitialize Values

Constants passed to RTPMPInitialize.

```
enum {
  kRTPMPRealtimeModeFlag        = 0x00000001
};
```

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

RTPMPIdle Values

Constants passed to RTPMPIdle.

```
enum {
  kRTPMPStillProcessingData     = 0x00000001 /* not done with data you've got*/
};
```

Declared In
QuickTimeStreaming.h, QTStreamingComponents.h

kRTPMPRespectDurationFlag

Constants grouped with kRTPMPRespectDurationFlag.

```
enum {
  kRTPMPSyncSampleFlag          = 0x00000001,
  kRTPMPRespectDurationFlag     = 0x00000002
};
```

Constants

`kRTPMPSyncSampleFlag`

The sample is a sync sample.

Available in Mac OS X v10.0 and later.

Declared in `QTStreamingComponents.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPRssmSetCapabilities Values

Constants passed to RTPRssmSetCapabilities.

```
enum {
  kRTPRssmEveryPacketAChunkFlag = 0x00000001,
  kRTPRssmQueueAndUseMarkerBitFlag = 0x00000002,
  kRTPRssmTrackLostPacketsFlag  = 0x00010000,
  kRTPRssmNoReorderingRequiredFlag = 0x00020000
};
```

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

RTPRssmSendPacketList Values

Constants passed to RTPRssmSendPacketList.

```
enum {
  kRTPRssmLostSomePackets       = 0x00000001
};
```

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

SHExtendedChunkRecord Values

Constants passed to SHExtendedChunkRecord.

```
enum {
  kSHExtendedChunkFlag_HasSampleCount = 1 << 0,
  kSHExtendedChunkFlag_HasFrameLengths = 1 << 1
};
```

Constants

`kSHExtendedChunkFlag_HasSampleCount`

Sample count data is added.

Available in Mac OS X v10.2 and later.

Declared in `QTStreamingComponents.h`.

Declared In

`QuickTimeStreaming.h, QTStreamingComponents.h`

Document Revision History

This table describes the changes to *QuickTime Streaming Reference*.

Date	Notes
2006-05-23	New document, based on previously published material, that describes the API for QuickTime Streaming.

Document Revision History

Index

R

S

T