



VIVOTEK NETWORK DEVELOPMENT PLATFORM

Audio/Video Reader Module

Version 1.0.0.1

2017/01/20

© 2017 VIVOTEK Inc. All Right Reserved

VIVOTEK may make changes to specifications and product descriptions at any time, without notice.

The following is trademarks of VIVOTEK Inc., and may be used to identify VIVOTEK products only: VIVOTEK. Other product and company names contained herein may be trademarks of their respective owners.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from VIVOTEK Inc.

TABLE OF CONTENTS

TABLE OF CONTENTS	1
1. OVERVIEW.....	2
1.1 Introduction	2
1.2 Getting Started with AVReader Module	2
1.3 File Structure	2
2. PROGRAMMER'S GUIDE	3
2.1 Using AVReader Module	3
3. SAMPLE CODE	4
3.1 Play 3GP File	4
3.2 Authenticate 3GP File.....	5
4. API REFERENCE	6
4.1 Enumeration.....	6
4.2 Data Structure	7
4.3 Callback Function	8
4.4 API Definition	9
4.4.1 AVReader_Initial	10
4.4.2 AVReader_InitialW.....	12
4.4.3 AVReader_Release.....	14
4.4.4 AVReader_SetAuthenticity	15
4.4.5 AVReader_OutputPacket.....	16
4.4.6 AVReader_GetMediaType	18

1. Overview

1.1 Introduction

This document describes the properties and methods supported by the VIVOTEK Audio Video Reader (AVReader) module.

1.2 Getting Started with AVReader Module

The main usage of AVReader module is to read a 3GP file which is recorded by AVComposer.

1.3 File Structure

Table 1-1 File Structure

File	Description
doc\VNDP_AVReader_API.pdf	This manual document
lib\d_AVReader.lib	The dynamic linking library
lib\AVReader.dll	The dynamic runtime library
inc\AVReader.h	Header file

2. Programmer's Guide

2.1 Using AVReader Module

- Start AVReader module.
- Retrieve frames from a 3GP file.

VIVOTEK CONFIDENTIAL
2017.01.20

3. Sample Code

3.1 Play 3GP File

Description

Read frames from a 3GP file.

Sample Code

- Step 1. Define an `AvSynchronizerStatusCallBack` function to receive status code, and define an `AvSynchronizerDisplayCallBack` function to receive information which carried in `TDISPLAYINFO`.
- Step 2. Initialize `AvSynchronizer`; pass `AvSynchronizerStatusCallBack` as the `LPSTATUSCALLBACK` callback function and `AvSynchronizerDisplayCallBack` as the `LPDISPLAYCALLBACK` callback function.
- Step 3. Create an `AVSynchronizer` playback channel.
- Step 4. Start the playback channel.
- Step 5. Define a `Read3GPFileThread` function to be executed by `Read3GPFileThread`. When `Read3GPFile` thread starts, use [AVReader Initial](#) to create an `AVReader` handle and open the specified 3GP file for reading.
- Step 6. Use [AVReader GetMediaType](#) to get media types that the file has. Then set media types to `AVSynchronizer` playback channel.
- Step 7. While receiving media packets through [AVReader OutputPacket](#), the thread passes these media packets to playback channel for display.
- Step 8. Until all frames are read and displayed, use [AVReader Release](#) to close 3GP file and delete `AVReader` handle.
- Step 9. Stop the playback channel.
- Step 10. Delete the playback channel and release `AvSynchronizer`.

3.2 Authenticate 3GP File

Description

Use AVComposer to save the frames from DataBroker to 3GP files with watermark; and use AVReader to verify them.

Sample Code

- Step 1. Initialize DataBroker.
- Step 2. Create connection.
- Step 3. Define a `GetNewFilename` function to generate specified media file name. Use **AVComposer_CreateHandle** to create AVComposer handle and pass `GetNewFilename` as the `AVCOMPOSER_PF_REQUEST_FILENAME` callback function. AVComposer will be ready to save input media packets after calling **AVComposer_StartRecording**.
- Step 4. Connect to a device, DataBroker uses asynchronous mechanism while connecting to device, the connection information would be carried in `DataBrokerStatusCallback` function.
- Step 5. While receiving media packets in `DataBrokerAVCallback`, it then uses **AVComposer_InputPacket** to save these media packets to audio or video track.
- Step 6. Besides to generate new file name in `GetNewFilename` function, application can use **AVComposer_SetAuthenticity** to set user specified password.
- Step 7. Disconnect device.
- Step 8. Delete connection.
- Step 9. Release DataBroker.
- Step 10. Use **AVComposer_StopRecording** to close 3GP file and **AVComposer_ReleaseHandle** to delete AVComposer handle.
- Step 11. Use [AVReader_Initial](#) to create an AVReader handle and open the specified 3GP file (`Authenticate3GPFile_0.3gp`) for reading.
- Step 12. Use [AVReader_SetAuthenticity](#) to set user specified password.
- Step 13. While receiving media packets through [AVReader_OutputPacket](#); AVReader checks watermark for each media packet in `Authenticate3GPFile_0.3gp`.
- Step 14. User can repeat step 11 to 13 for checking other 3GP files.

4. API Reference

This chapter describes the API function calls for the AVReader module.

4.1 Enumeration

AVReader has no enumeration definitions.

VIVOTEK CONFIDENTIAL
2017.01.20

4.2 Data Structure

AVReader has no data structure definitions.

VIVOTEK CONFIDENTIAL
2017.01.20

4.3 Callback Function

AVReader has no callback function definitions.

VIVOTEK CONFIDENTIAL
2017.01.20

4.4 API Definition

The API definition is depicted here.

VIVOTEK CONFIDENTIAL
2017.01.20

4.4.1 AVReader_Initial

This function creates the AVReader object and returns an interface to it. You must call this function or [AVReader_InitialW](#) before using this module.

Syntax

```
SCODE AVReader_Initial (  
  
                                AVReader                **ppAVReader,  
  
                                char                    *pszFilename,  
  
                                BOOL                     bDiscardVivo  
  
);
```

Parameters

****ppAVReader**

[out] Pointer to an AVReader interface.

pszFilename

[in] Filename.

bDiscardVivo

[in] Ignore VIVOTEK proprietary data or not.

Return Values

S_OK

Create the module successfully.

S_FAIL

Fail to create the module.

Remarks

Applications that need to use information in proprietary data for special integration should set bDiscardVivo to FALSE.

Requirements

AVReader.h

[See Also](#)

VIVOTEK CONFIDENTIAL
2017.01.20

4.4.2 AVReader_InitialW

This function (with Unicode filename) creates the AVReader object and returns an interface to it. You must call this function or [AVReader_Initial](#) before using this module.

Syntax

```
SCODE AVReader_InitialW (  
  
                                AVReader                **ppAVReader,  
  
                                wchar_t                  *pszFilename,  
  
                                BOOL                      bDiscardVivo  
  
);
```

Parameters

****ppAVReader**

[out] Pointer to an AVReader interface.

pszFilename

[in] Filename.

bDiscardVivo

[in] Ignore VIVOTEK proprietary data or not.

Return Values

S_OK

Create the module successfully.

S_FAIL

Fail to create the module.

Remarks

Applications that need to use information in proprietary data for special integration should set bDiscardVivo to FALSE.

Requirements

AVReader.h

[See Also](#)

VIVOTEK CONFIDENTIAL
2017.01.20

4.4.3 AVReader_Release

Call this function to release the AVReader object.

Syntax

```
SCODE AVReader_Release (  
  
                                AVReader                *pAVReader  
  
);
```

Parameters

***pAVReader**

[in] An AVReader interface which is returned by [AVReader_Initial](#) or [AVReader_InitialW](#).

.

Return Values

S_OK

Release the object successfully.

S_FAIL

Fail to release the object.

Remarks

Requirements

AVReader.h

See Also

4.4.4 AVReader_SetAuthenticity

Set authenticity password.

Syntax

```
void AVReader_SetAuthenticity (
                                AVReader          *pAVReader,
                                char                *pszPassword
);
```

Parameters

***pAVReader**

[in] An AVReader interface which is returned by [AVReader_Initial](#) or [AVReader_InitialW](#).

***pszPassword**

[in] Password. This is used while verifying VIVOTEK proprietary watermark.

Return Values

Remarks

Requirements

AVReader.h

See Also

4.4.5 AVReader_OutputPacket

Read a media packet from file.

Syntax

```
SCODE AVReader_OutputPacket (
    AVReader                    *pAVReader,
    TMediaDataPacketInfoV3     **pptMediaPacket
);
```

Parameters

***pAVReader**

[in] An AVReader interface which is returned by [AVReader_Initial](#) or [AVReader_InitialW](#).

****pptMediaPacket**

[out] A Pointer to a media packet's start address.

Return Values

S_OK

Read an output packet successfully. And authentication is approved while authenticity has been set.

S_FAIL

Fail to read the output packet. Application should use [AVReader_Release](#) to delete AVReader object while get this error code.

S_AVREADER_EOF

End of this file. Application should use [AVReader_Release](#) to release AVReader while get this error code.

S_AVREADER_NO_PROPER_AUTH

Read an output packet successfully. But this packet has no authentication info.

S_AVREADER_AUTH_NOT_MATCH

Read an output packet successfully, but fail to pass the authentication.

Remarks

Requirements

AVReader.h

See Also

VIVOTEK CONFIDENTIAL
2017.01.20

4.4.6 AVReader_GetMediaType

Get one or more media types in an opened 3GP file.

Syntax

```
DWORD AVReader_GetMediaType (  
                                AVReader                                *pAVReader  
);
```

Parameters

***pAVReader**

[in] An AVReader interface which is returned by [AVReader_Initial](#) or [AVReader_InitialW](#).

Return Values

The return value is a combination of the TsdMediaTypes values which are defined in SrvTypeDef.h.

Remarks

Requirements

AVReader.h

See Also
