



HiVS

API Reference

Issue 05

Date 2018-07-18

Copyright © HiSilicon (Shanghai) Technologies Co., Ltd. 2019. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon (Shanghai) Technologies Co., Ltd.

Trademarks and Permissions



HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon (Shanghai) Technologies Co., Ltd.

Address: New R&D Center, 49 Wuhe Road, Bantian,
Longgang District,
Shenzhen 518129 P. R. China

Website: <http://www.hisilicon.com/en/>

Email: support@hisilicon.com



About This Document

Purpose

This document provides reference information including the application programming interfaces (APIs), header files, and error codes for the programmers that develop products or solutions using the intelligent video surveillance (IVS) module of HiSilicon media processors.



NOTE

This document uses Hi3518E V200 as an example. Unless otherwise specified, this document applies to Hi3518E V200, Hi3518E V201, Hi3521A, Hi3520D V300, Hi3531A, Hi3536C V100, Hi3531D V100, Hi3521D V100, Hi3520D V400, Hi3556A V100, Hi3519A V100, Hi3516E V200, Hi3516D V200, Hi3516E V300, and Hi3518E V300.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3518E	V200
Hi3518E	V201
Hi3516C	V200
Hi3521A	V100
Hi3520D	V300
Hi3531A	V100
Hi3519	V100
Hi3519	V101
Hi3516A	V200
Hi3516C	V300
Hi3516E	V100
Hi3559A	V100ES
Hi3536C	V100
Hi3559A	V100



Product Name	Version
Hi3559C	V100
Hi3531D	V100
Hi3521D	V100
Hi3520D	V400
Hi3519A	V100
Hi3556A	V100
Hi3516C	V500
Hi3516D	V300
Hi3516A	V300
Hi3559	V200
Hi3556	V200
Hi3516E	V200
Hi3516E	V300
Hi3518E	V300
Hi3516D	V200




Intended Audience

This document is intended for:



- Technical support engineers
- Software development engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.



Symbol	Description
	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
 NOTE	Calls attention to important information, best practices and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Changes between document issues are cumulative. The latest document issue contains all changes made in previous issues.

Issue 05 (2018-07-18)

This issue is the fifth official release.

Issue 04 (2017-11-15)

This issue is the fourth official release, which incorporates the following changes:

The description of the Hi3559A V100 is added.

Issue 03 (2017-10-23)

This issue is the third official release, which incorporates the following changes:

Chapter 2, the description in the **Syntax** field of HI_IVS_MD_GetBg is modified.

Issue 02 (2017-03-28)

This issue is the second official release, which incorporates the following changes:

The descriptions related to the Hi3559A V100ES and Hi3536C are added.

Issue 01 (2017-02-15)

This issue is the first official release, which incorporates the following changes:

Chapter 3 Data Structures

The description in the **Member** field of MD_ATTR_S is modified.

Issue 00B05 (2016-05-10)

This issue is the fifth draft release, which incorporates the following changes:

The contents related to the Hi3519 V101 and Hi3516CV300 are added.



Issue 00B04 (2015-12-15)

This issue is the fourth draft release, which incorporates the following changes:

Chapter 2 API Reference

The descriptions in the **Syntax** and **Parameter** fields of HI_IVS_MD_Process are modified.

Chapter 4 Error Codes

Table 4-1 is modified.

Chapter 3 Data Structures

The description in the **Member** field of MD_ATTR_S is modified.

Chapter 5 Proc Debugging Information

Section 5.2 is modified.

Issue 00B03 (2015-09-20)

This issue is the third draft release, which incorporates the following changes:

Chapter 3 Data Structures

The description in the **Member** field of MD_ATTR_S is modified.

Chapter 5 Proc Debugging Information

This chapter is added.

Issue 00B02 (2015-07-29)

This issue is the second draft release, which incorporates the following changes:

The contents related to the Hi3531A are added.

Issue 00B01 (2015-04-10)

This issue is the first draft release.



Contents

1 Introduction to the IVS	1
1.1 Overview	1
2 MD	2
2.1 Function Description	2
2.1.1 Motion Detection	2
2.2 API Reference	3
2.3 Data Structures.....	11
2.4 Error Codes.....	14
2.5 Proc Debugging Information	16
2.5.1 Overview	16
2.5.2 MD Proc Information.....	16



1 Introduction to the IVS

1.1 Overview

The intelligent video surveillance (IVS) module provides optimized intelligent video surveillance for HiSilicon media processors compared with the intelligent video engine (IVE). You can rapidly develop relevant intelligent applications based on the IVS. Currently, the IVS module supports motion detection (MD) .



2 MD

2.1 Function Description

2.1.1 Motion Detection

The MD application is used to detect the motion status of the video by detecting the video luminance variance, and obtain the video detection analysis (VDA) result.

Basic Concepts

The basic concepts of the MD application are as follows:

- MD algorithm

The MD algorithms include the frame difference algorithm (MD_ALG_MODE_REF) and background algorithm (MD_ALG_MODE_BG).

- Frame difference algorithm

This algorithm uses a user-specified picture as the reference frame to obtain the VDA result.

- Background algorithm

The background picture of the current video is generated during MD processing. This algorithm uses the background picture as the reference frame to obtain the VDA result.

- Background update weight

If the background algorithm is used, still pictures are abstracted during MD processing. The pixel value of the still picture and that of the background picture are blended. The formula is as follows:

Pixel value of the new background = (Blending weight of the still picture x Pixel value of the still picture + Blending weight of the dynamic picture x Pixel value of the old background) >> 16

The blending weight of the still picture and that of the dynamic picture are **u0q16X** and **u0q16Y**, respectively.



NOTICE

For Hi3559A V100ES/Hi3559A V100, the required MMZ address must be in a 4 GB space when the 64-bit OS is used. Otherwise, exceptions occur.

2.2 API Reference

The MD application provides the following APIs for initializing and exiting the MD application, obtaining and releasing the handle, obtaining the background, and implementing detection:

- [HI_IVS_MD_Init](#): Initializes the MD application.
- [HI_IVS_MD_Exit](#): Exits the MD application.
- [HI_IVS_MD_CreateChn](#): Creates an MD channel.
- [HI_IVS_MD_DestroyChn](#): Destroys the MD channel.
- [HI_IVS_MD_SetChnAttr](#): Sets the attributes of an MD channel.
- [HI_IVS_MD_GetChnAttr](#): Obtains the attributes of an MD channel.
- [HI_IVS_MD_GetBg](#): Obtains the MD background.
- [HI_IVS_MD_Process](#): Implements the MD.

HI_IVS_MD_Init

[Description]

Initializes the MD application.

[Syntax]

```
HI_S32 HI_IVS_MD_Init (HI_VOID) ;
```

[Parameter]

None

[Return Value]

Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]



- Call this API to initialize the MD before calling other APIs. Note that this API can be called only once. If this API is called repeatedly, an error code is returned.
- This API must work with [HI_IVS_MD_Exit](#).

[Example]

None

[See Also]

[HI_IVS_MD_Exit](#)

HI_IVS_MD_Exit

[Description]

Exits the MD application.

[Syntax]

```
HI_S32 HI_IVS_MD_Exit (HI_VOID) ;
```

[Parameter]

None

[Return Value]

Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]

Call [HI_IVS_MD_Init](#) to initialize the MD before calling this API. Otherwise, an error code is returned.

[Example]

None

[See Also]

[HI_IVS_MD_Init](#)

HI_IVS_MD_CreateChn

[Description]

Creates an MD channel.



[Syntax]

```
HI_S32 HI_IVS_MD_CreateChn(MD_CHN MdChn, MD_ATTR_S *pstMdAttr);
```

[Parameter]

Parameter	Description	Input/Output
MdChn	Channel ID Value range: [0, 63]	Input
pstMdAttr	Pointer to the channel information It cannot be null.	Input

[Return Value]

Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]

Call [HI_IVS_MD_Init](#) to initialize the MD before calling this API. Otherwise, an error code is returned.

[Example]

None

[See Also]

- [HI_IVS_MD_DestroyChn](#)
- [HI_IVS_MD_SetChnAttr](#)
- [HI_IVS_MD_GetChnAttr](#)
- [HI_IVS_MD_GetBg](#)
- [HI_IVS_MD_Process](#)

HI_IVS_MD_DestroyChn

[Description]

Destroys the MD channel.

[Syntax]

```
HI_S32 HI_IVS_MD_DestroyChn(MD_CHN MdChn);
```



[Parameter]

Parameter	Description	Input/Output
MdChn	Channel ID Value range: [0, 63]	Input

[Return Value]

Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]

- Call [HI_IVS_MD_Init](#) to initialize the MD before calling this API. Otherwise, an error code is returned.
- **MdChn** must be the ID of a channel created by calling [HI_IVS_MD_CreateChn](#). Otherwise, an error code is returned.

[Example]

None

[See Also]

- [HI_IVS_MD_CreateChn](#)
- [HI_IVS_MD_SetChnAttr](#)
- [HI_IVS_MD_GetChnAttr](#)
- [HI_IVS_MD_GetBg](#)
- [HI_IVS_MD_Process](#)

HI_IVS_MD_SetChnAttr

[Description]

Sets the attributes of an MD channel.

[Syntax]

```
HI_S32 HI_IVS_MD_SetChnAttr(MD_CHN MdChn, MD_ATTR_S *pstMdAttr);
```

[Parameter]



Parameter	Description	Input/Output
MdChn	Channel ID Value range: [0, 63]	Input
pstMdAttr	Pointer to the channel information It cannot be null.	Input

[Return Value]

Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]

- Call [HI_IVS_MD_Init](#) to initialize the MD before calling this API. Otherwise, an error code is returned.
- **MdChn** must be the ID of a channel created by calling [HI_IVS_MD_CreateChn](#). Otherwise, an error code is returned.
- The static attributes (**enAlgMode**, **enSadMode**, **u16Width**, and **u16Height**) of the channel cannot be modified. Otherwise, an error code is returned.

[Example]

None

[See Also]

- [HI_IVS_MD_CreateChn](#)
- [HI_IVS_MD_DestroyChn](#)
- [HI_IVS_MD_GetChnAttr](#)
- [HI_IVS_MD_GetBg](#)
- [HI_IVS_MD_Process](#)

HI_IVS_MD_GetChnAttr

[Description]

Obtains the attributes of an MD channel.

[Syntax]

```
HI_S32 HI_IVS_MD_GetChnAttr(MD_CHN MdChn, MD_ATTR_S *pstMdAttr);
```



[Parameter]

Parameter	Description	Input/Output
MdChn	Channel ID Value range: [0, 63]	Input
pstMdAttr	Pointer to the channel information It cannot be null.	Output

[Return Value]

Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]

- Call [HI_IVS_MD_Init](#) to initialize the MD before calling this API. Otherwise, an error code is returned.
- **MdChn** must be the ID of a channel created by calling [HI_IVS_MD_CreateChn](#). Otherwise, an error code is returned.

[Example]

None

[See Also]

- [HI_IVS_MD_CreateChn](#)
- [HI_IVS_MD_DestroyChn](#)
- [HI_IVS_MD_SetChnAttr](#)
- [HI_IVS_MD_GetBg](#)
- [HI_IVS_MD_Process](#)

HI_IVS_MD_GetBg

[Description]

Obtains the MD background.

[Syntax]

```
HI_S32 HI_IVS_MD_GetBg(MD_CHN MdChn, IVE_DST_IMAGE_S *pstBg);
```



[Parameter]

Parameter	Description	Input/Output
MdChn	Channel ID Value range: [0, 63]	Input
pstBg	Pointer to the background picture It cannot be null.	Output

Parameter	Supported Picture Type	Address Alignment Mode	Resolution
pstBg	U8C1	16 bytes	64 x 64 to 1920 x 1080

[Return Value]

Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]

- Call [HI_IVS_MD_Init](#) to initialize the MD before calling this API. Otherwise, an error code is returned.
- **MdChn** must be the ID of a channel created by calling [HI_IVS_MD_CreateChn](#). Otherwise, an error code is returned.
- The background data can be obtained only when the background algorithm is used. Otherwise, an error code is returned when this API is called.

[Example]

None

[See Also]

- [HI_IVS_MD_CreateChn](#)
- [HI_IVS_MD_DestroyChn](#)
- [HI_IVS_MD_SetChnAttr](#)
- [HI_IVS_MD_GetBg](#)
- [HI_IVS_MD_Process](#)



HI_IVS_MD_Process

[Description]

Implements the MD.

[Syntax]

```
HI_S32 HI_IVS_MD_Process(MD_CHN MdChn, IVE_SRC_IMAGE_S *pstCur,  
IVE_SRC_IMAGE_S *pstRef, IVE_DST_IMAGE_S *pstSad, IVE_DST_MEM_INFO_S  
*pstBlob);
```

[Parameter]

Parameter	Description	Input/Output
MdChn	Channel ID Value range: [0, 63]	Input
pstCur	Pointer to the current frame picture It cannot be null.	Input
pstRef	Pointer to the reference frame picture It cannot be null.	Input
pstSad	SAD pointer According to pstMdAttr → enSadOutCtrl , pstSad cannot be null if the output is required.	Output
pstBlob	Pointer to the region information It cannot be null.	Output

Parameter	Supported Picture Type	Address Alignment Mode	Resolution
pstCur	U8C1	16 bytes	64 x 64 to 1920 x 1080
pstRef	U8C1	16 bytes	64 x 64 to 1920 x 1080
pstSad	U8C1/U16C1	16byte	According to pstMdAttr → enSadMode , the height and width are 1/4, 1/8, or 1/16 of the height and width of pstCur in 4x4, 8x8, or 16x16 blocking mode respectively.
pstBlob	-	16 bytes	-

[Return Value]



Return Value	Description
0	Success
Other values	Failure. An error code is returned. For details, see section 2.4 "Error Codes."

[Requirement]

- Header files: **hi_comm_ive.h**, **hi_md.h**, and **ivs_md.h**
- Library file: **libmd.a** (ivs_md.lib used on the PC for simulation)

[Note]

- Call [HI_IVS_MD_Init](#) to initialize the MD before calling this API. Otherwise, an error code is returned.
- **MdChn** must be the ID of a channel created by calling [HI_IVS_MD_CreateChn](#). Otherwise, an error code is returned.
- A maximum of 254 pieces of region information can be output. For details, see IVE_CCBLOB_S in chapter 3 "Data Structures" of the *HiIVE API Reference*. The member **u16CurAreaThr** in IVE_CCBLOB_S indicates the information of the area threshold after blocking. The connected component information output here is stored consecutively.

[Example]

None

[See Also]

- [HI_IVS_MD_CreateChn](#)
- [HI_IVS_MD_DestroyChn](#)
- [HI_IVS_MD_SetChnAttr](#)
- [HI_IVS_MD_GetBg](#)
- [HI_IVS_MD_Process](#)

2.3 Data Structures

The MD provides the following data structures:

- [MD_ALG_MODE_E](#): Defines the MD algorithms.
- [MD_ATTR_S](#): Defines the attributes of an MD channel.

MD_ALG_MODE_E

[Description]

Defines the MD algorithms.

[Syntax]

```
typedef enum hiMD_ALG_MODE_E
```



```
{  
    MD_ALG_MODE_BG = 0x0, /* Base on background image*/  
    MD_ALG_MODE_REF = 0x1, /* Base on reference image*/  
    MD_ALG_MODE_BUTT  
}MD_ALG_MODE_E;
```

[Member]

Member	Description
MD_ALG_MODE_BG	Background algorithm
MD_ALG_MODE_REF	Frame difference algorithm

[Note]

None

[See Also]

None

MD_ATTR_S

[Description]

Defines the attributes of an MD channel.

[Syntax]

```
typedef struct hiMD_ATTR_S  
{  
    MD_ALG_MODE_E      enAlgMode; /* Md algorithm mode*/  
    IVE_SAD_MODE_E      enSadMode; /* Sad mode*/  
    IVE_SAD_OUT_CTRL_E  enSadOutCtrl; /*Sad output ctrl*/  
    HI_U16              ul6Width; /* Image width*/  
    HI_U16              ul6Height; /* Image height*/  
    HI_U16              ul6SadThr; /* Sad thresh*/  
    IVE_CCL_CTRL_S      stCclCtrl; /* Ccl ctrl*/  
    IVE_ADD_CTRL_S      stAddCtrl; /* Add ctrl*/  
}MD_ATTR_S
```

For Hi3559A V100ES/Hi3559A V100:

```
typedef struct hiMD_ATTR_S  
{  
    MD_ALG_MODE_E      enAlgMode; /*Md algorithm mode*/  
    IVE_SAD_MODE_E      enSadMode; /*Sad mode*/  
    IVE_SAD_OUT_CTRL_E  enSadOutCtrl; /*Sad output ctrl*/  
    HI_U32              u32Width; /*Image width*/  
    HI_U32              u32Height; /*Image height*/  
}
```



```
HI_U16          u16SadThr; /*Sad thresh*/  
IVE_CCL_CTRL_S  stCclCtrl; /*Ccl ctrl*/  
IVE_ADD_CTRL_S  stAddCtrl; /*Add ctrl*/  
}MD_ATTR_S;
```

[Member]

Member	Description
enAlgMode	Algorithm. For details, see MD_ALG_MODE_E .
enSadMode	Sum of absolute difference (SAD) mode. For details, see IVE_SAD_MODE_E in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> .
enSadOutCtrl	Sad output control. For details, see IVE_SAD_OUT_CTRL_E in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> . Only the output control of IVE_SAD_OUT_CTRL_16BIT_BOTH, IVE_SAD_OUT_CTRL_8BIT_BOTH, and IVE_SAD_OUT_CTRL_THRESH is supported.
u16Width	Picture width. The value must be an integral multiple of the macroblock width. The value range is [64, 1920].
u16Height	Picture height. The value must be an integral multiple of the macroblock height. The value range is [64, 1080].
u16SadThr	SAD threshold. The value depends on the enSadOutCtrl . <ul style="list-style-type: none">For IVE_SAD_OUT_CTRL_8BIT_BOTH Value range: [0, 255]For IVE_SAD_OUT_CTRL_16BIT_BOTH and IVE_SAD_OUT_CTRL_THRESH Value range: [0, 65535]
stCclCtrl	Connected component labeling (CCL) control parameter. For details, see IVE_CCL_CTRL_S in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> . The member information of the CCL control parameter takes effect on the picture after blocking.
stAddCtrl	Addition control parameter. For details, see IVE_ADD_CTRL_S in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> .

For Hi3559A V100ES/Hi3559A V100:

Member	Description
enAlgMode	Algorithm. For details, see MD_ALG_MODE_E
enSadMode	Sum of absolute difference (SAD) mode. For details, see IVE_SAD_MODE_E in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> .



Member	Description
enSadOutCtrl	Sad output control. For details, see IVE_SAD_OUT_CTRL_E in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> . Only the output control of IVE_SAD_OUT_CTRL_16BIT_BOTH, IVE_SAD_OUT_CTRL_8BIT_BOTH, and IVE_SAD_OUT_CTRL_THRESH is supported.
u32Width	Picture width. The value must be an integral multiple of the macroblock width. The value range is [64, 1920].
u32Height	Picture height. The value must be an integral multiple of the macroblock height. The value range is [64, 1080].
u16SadThr	SAD threshold. The value depends on the enSadOutCtrl . For IVE_SAD_OUT_CTRL_8BIT_BOTH Value range: [0, 255] For IVE_SAD_OUT_CTRL_16BIT_BOTH and IVE_SAD_OUT_CTRL_THRESH Value range: [0, 65535]
stCclCtrl	Connected component labeling (CCL) control parameter. For details, see IVE_CCL_CTRL_S in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> . The member information of the CCL control parameter takes effect on the picture after blocking.
stAddCtrl	Addition control parameter. For details, see IVE_ADD_CTRL_S in chapter 3 "Data Structures" of the <i>HiIVE API Reference</i> .

[Note]

None

[See Also]

None

2.4 Error Codes

Table 2-1 lists the error codes of IVS APIs, most of which are the same as the error codes of IVE APIs. The error codes of the IVS APIs that are different from those of the IVE APIs are listed in the latter part of table.

Table 2-1 Error codes of IVS APIs

Error Code	Macro Definition	Description
0xA01D8001	HI_ERR_IVE_INVALID_DEVID	The device ID is invalid.



Error Code	Macro Definition	Description
0xA01D8002	HI_ERR_IVE_INVALID_CHNID	The channel group ID or the region handle is invalid.
0xA01D8003	HI_ERR_IVE_ILLEGAL_PARAM	The parameter is invalid.
0xA01D8004	HI_ERR_IVE_EXIST	The device, channel, or resource to be created already exists.
0xA01D8005	HI_ERR_IVE_UNEXIST	The device, channel, or resource to be used or destroyed does not exist.
0xA01D8006	HI_ERR_IVE_NULL_PTR	The pointer is null.
0xA01D8007	HI_ERR_IVE_NOT_CONFIG	The module is not configured.
0xA01D8008	HI_ERR_IVE_NOT_SUPPORT	The parameter or function is not supported.
0xA01D8009	HI_ERR_IVE_NOT_PERM	The operation, for example, modifying the value of a static parameter, is forbidden.
0xA01D800C	HI_ERR_IVE_NOMEM	The memory fails to be allocated for the reasons such as system memory insufficiency.
0xA01D800D	HI_ERR_IVE_NOBUF	The buffer fails to be allocated. The reason may be that the requested picture buffer is too large.
0xA01D800E	HI_ERR_IVE_BUF_EMPTY	There is no picture in the buffer.
0xA01D800F	HI_ERR_IVE_BUF_FULL	The buffer is full of pictures.
0xA01D8010	HI_ERR_IVE_NOTREADY	The system is not initialized or the corresponding module driver is not loaded.
0xA01D8011	HI_ERR_IVE_BADADDR	The address is invalid.
0xA01D8012	HI_ERR_IVE_BUSY	The system is busy.
0xA01D8040	HI_ERR_IVE_SYS_TIMEOUT	The IVE times out.
0xA01D8041	HI_ERR_IVE_QUERY_TIMEOUT	The query times out.
0xA01D8042	HI_ERR_IVE_OPEN_FILE	Opening a file fails.
0xA01D8043	HI_ERR_IVE_READ_FILE	Reading a file fails.
0xA01D8044	HI_ERR_IVE_WRITE_FILE	Writing to a file fails.
0xA0308002	HI_ERR_ODT_INVALID_CHNID	The on-die termination (ODT) channel group ID or the region handle is invalid.



Error Code	Macro Definition	Description
0xA0308004	HI_ERR_ODT_EXIST	The device, channel, or resource to be created already exists.
0xA0308005	HI_ERR_ODT_UNEXIST	The device, channel, or resource to be used or destroyed does not exist.
0xA0308009	HI_ERR_ODT_NOT_PERM	The operation, for example, modifying the value of a static parameter, is forbidden.
0xA0308010	HI_ERR_ODT_NOTREADY	The ODT is not initialized.
0xA0308012	HI_ERR_ODT_BUSY	The ODT is busy.

2.5 Proc Debugging Information

2.5.1 Overview

The debugging information is obtained from the proc file system on Linux. The information reflects the current system status and can be used to locate and analyze problems.

[File Directory]

/proc/umap

View the proc information in either of the following ways:

- Run a **cat** command such as **cat /proc/umap/md** on the console to view the proc information about MD, or run file operation commands such as **cp /proc/umap/md ./** to copy files to the current directory.
- Read the preceding files as common read-only files through applications such as **fopen** and **fread**.



NOTE

Note the following when reading parameter descriptions:

- For the parameter whose value is **0** or **1**, if mapping between the values and definitions is not specified, the value **1** indicates affirmative and the value **0** indicates negative.
- For the parameter whose value is **aaa**, **bbb**, or **ccc**, if the mapping between the values and the definitions is not specified, identify the parameter definitions based on **aaa**, **bbb**, or **ccc**.

2.5.2 MD Proc Information

[Debugging Information]

```
~ # cat /proc/umap/md
```

```
[MD] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 27 2015, 17:05:44]
```

```
-----MD CHN ATTR-----
```

```
NO.    W    H    Alg    SadMode    SadOutCtrl    SadT    CclMode    CclInitT    CclStep
```



0	720	576	0	0	0	200	1	16	4
XWt	YWt	FrmRate	CostTmPerFrm						
32768	32768	19	72990						

[Analysis]

This section records the working status information about MD.

[Parameter Description]

Parameter		Description
MD CHN ATTR	NO.	Channel ID
	W	Channel width (in pixel)
	H	Channel height (in pixel)
	Alg	Working algorithm 0: background algorithm 1: frame difference algorithm
	SadMode	SAD mode 0: 4 x 4 macroblocks 1: 8 x 8 macroblocks 2: 16 x 16 macroblocks
	SadOutCtrl	Sad output control 0: IVE_SAD_OUT_CTRL_16BIT_BOTH 1: IVE_SAD_OUT_CTRL_8BIT_BOTH 4: IVE_SAD_OUT_CTRL_THRESH
	SadT	SAD threshold
	CclMode	CCL mode 0: 4-connected-component 1: 8-connected-component
	CclInitT	Initial CCL threshold
	CclStep	CCL step
	XWt	X weight for background algorithm update
	YWt	Y weight for background algorithm update
	FrmRate	Frame rate
	CostTmPerFrm	Time consumption of each frame (unit: μ s)

[Note]

- For some chips, the configuration of **CclMode** cannot be modified. For details, see the *HiIVE API Reference*.



- The frame rate and time consumption of each frame are collected for statistics every 10 seconds.