

Iris ID

iCAM T10 SDK User Manual

Document ver 1.01 – Device Control API

Copyright © 2012 Iris ID Systems, Inc. - All rights reserved.

iCAM T10 SDK User Manual

If this manual is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this manual may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Iris ID Systems, Inc. Please note that the content in this manual is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Iris ID (formerly LG IRIS). Iris ID Systems, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this manual.

Please remember that existing images or drawings that you may want to include in your document may be protected under copyright law. The unauthorized incorporation of such material into your work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Iris ID, Iris ID logo, LG IRIS, LG IRIS logo and IrisAccess® are either registered trademarks or trademarks of Iris ID Systems, Inc. in the United States and/or other countries.

All product names mentioned in this document are trademarks, registered trademarks, or copyrights of their respective holders.

Iris ID Systems, Inc. 7 Clarke Drive, Cranbury, New Jersey 08512, USA.

Document No. IRISID71SDK-01-0901-0913

Last Update: March, 2013

Table of Contents

1. Introduction	5
2. Purpose	5
3. Scope	6
4. Definitions, Acronyms, and Abbreviations	6
5. iCAM T10 SDK Sample Application	6
5.1 Details of the iCAM T10 Sample Application	6
5.2 Minimum Requirements	7
5.3 Software Installation	7
5.4 Installing the iCAM T10 USB Driver	7
5.5 Verifying the iCAM T10 Device driver and manual driver installation	8
5.6 Opening a Sample Application	16
5.7 Using the iCAM T10 Sample Application	17
5.7.1 Steps to Capture Iris Enrollment Image (standard Steps)	17
5.7.2 Steps to Recognize a User	19
5.8 How to Operate iCAM T10	19
5.8.1 Operation	19
5.8.2 iCAM T10 SDK Sample Application Options & Features	20
6. iCAM T10 SDK Device Control	22
6.1 Overview	22
6.2 Software Development Compatibility	22
6.3 Software Architecture	22
6.4 General Procedures for Capturing Images	23
6.5 Application Programming Interface Function & Event Summary	26
6.6 Application Programming Interface	28
6.6.1 t10GetVersion()	28
6.6.2 t10Open()	29
6.6.3 t10Close()	30
6.6.4 t10GetSerialNumber()	31
6.6.5 t10SetMoveIrisPosition()	32
6.6.6 t10SetBeeper()	34
6.6.7 t10SetIllumination()	35
6.6.8 t10StartIrisCapture()	36
6.6.9 t10CaptureIris()	38
6.6.10 t10AbortIrisCapture()	39
6.6.11 T10SetCallback()	40
6.7 Events	41
6.7.1 Fire_OnGetStatus	41
6.7.2 Fire_OnGetLiveInfo	42
6.7.3 Fire_OnGetIrisImage	43
6.8 Constants	45
7. Appendix	47

7.1	Device Control Definitions.....	47
7.1.1	iCAM T10 Device Error Codes.....	47
8.	Reference	47
9.	Technical Support	47

1. Introduction

iCAM T10 SDK

iCAM T10 SDK provides the application programming interfaces in Microsoft Windows OS environment to control the iCAM T10 USB Iris Camera

In order to use the SDK, the Pentium compatible PC must have a USB 2.0 (or above) interface to connect to iCAM T10 device.

iCAM T10 SDK contains the device control functions to control camera unit, for example, taking iris image(s) for enrollment, and for recognition, etc.

iCAM T10 Device

The iCAM T10 is designed for use as an iris enrollment and recognition camera unit using a single USB cable for connection and power. With the Visor, users can quickly and easily gauge proper usage positioning, making it possible to overcome external light and usage issues.

The incorporated orientation sensor prevents misuse of the product (such as upside down (rotating 180 degrees)).

Dimension (W x H x D)	6.55" x 6.92" x 2.60" (166.5mm x 176mm x 66.2mm)
Weight	0.96lbs (438g / Camera Unit: 0.58lbs, Sun Visor: 0.38lbs)
Power Input	USB Bus Powered
Iris Camera	1.2MP Auto & Manual Dual Iris Capture with B&W Image Sensor
Iris Capture Range	4.92" (125mm)
Interface	High Speed USB 2.0
Indication	Green LED for Power Indication & Green LED for Right Eye Indication Beep Sound (optional by application)
Usability	Detachable Visor for cleaning the front window
Operating Temperature	32°F~122°F (0°C~50°C)
Storage Temperature	-4°F ~ 203°F (-20°C ~ 95°C)
Humidity	0 ~ 95% non-condensing
Standards	CE, FCC, Eye Safety IP54 for Dust and water protection

2. Purpose

This document identifies the requirements of interface functions for iCAM T10 SDK, which shall provide a convenient programming model for developing a T10 based iris biometric application by various vendors of biometric solution providers which can use C/C++ based Win32 dll.

The objective of this document is to act as the sole, authoritative and controlled source of technical requirements to be implemented in the iCAM T10 SDK.

** Note: Additional reference, amendments and updated documentation material may become available directly from the <http://www.irisid.com> website. Check the site for updated information, frequently asked questions, and tips to be used with your product.*

3. Scope

This document consolidates and categorizes the design of API's for an iCAM T10 SDK interface.

4. Definitions, Acronyms, and Abbreviations

- **iCAM** – A registered trademark name for iris cameras developed by Iris ID.
- **iCAM T10** – USB tethered type dual iris cameras developed by Iris ID.
- **COM** – Component Object Model
- **SDK**- Software Development Kit
- **Enrollment Image** – The image(s) captured during the enrollment process.
- **Recognition Image** - The image(s) captured during the recognition process.
- **Application Event Functions** – Functions that are used in an application that define or determine an event to occur.
- **Application Programming Interface (API)** - A specification intended to be used as an interface by software components to communicate with each other. These API's may include specifications for routines, data structures, object classes, and variables
- **Functions** - A block of code that has a name and a property that is re-usable. A self contained block of statements that perform a coherent task of the same kind.

5. iCAM T10 SDK Sample Application

5.1 Details of the iCAM T10 Sample Application

The iCAM T10 SDK sample application is located in `\Program files (x86)\Iris ID\iCAM T10 SDK\Samples\` by default. This program consists of two parts; 1) Enrollment iris images through an iCAM T10 unit, and 2) Recognition iris image capture. Saved images of either enrollment or recognition images can be viewed from the PC. This sample application serves as a good example to better understand how to manage information related to iris images and camera controls.

If you enroll or identify a user with the sample application, it will save all iris images to `C:\users\[USER ID]\Documents\iCAM T10 SDK\`.

**Note:* To clean up enrollment information in the sample program, please remove all of the files in `C:\users\[USER ID]\Documents\iCAM T10 SDK`

5.2 Minimum Requirements

Items Required for Use of This Product

Required Equipment (not provided by Iris ID):

- Windows based PC (Windows XP (32-bit) / Windows 7 (32-bit) / Windows 7 (64-bit))

Minimum Computer requirements:

- Microsoft Windows XP (32-bit), Windows 7 (32-bit), or Windows 7 (64-bit) OS
- 512MB RAM (or higher)
- x86 Processor, 2.0 GHz (or higher)
- 2GB available HDD space or above
- Microsoft .NET Framework ver. 3.5
- Mouse, SVGA Monitor, Keyboard
- Dedicated USB 2.0 port

***Note:** An available and dedicated USB 2.0 compliant port is needed to properly use the iCAM T10.

5.3 Software Installation

Steps for software installation

* Note: BEFORE CONNECTING iCAM T10 to your PC: Install iCAM T10 SDK software.

1. Select iCAM T10 SDK software (*select the appropriate x86 or x64 SDK build as needed*).
2. Open the Setup folder.
3. Select the Setup.exe Setup Launcher icon.
4. Select > Next at the Welcome to the InstallShield Wizard for iCAM T10 SDK.
5. Select radio button for License Agreement and press > Next.
6. Select destination folder to install and press > Next.
7. Select > "Install" to install software files.
8. Select > Finish to complete installation of software successfully.
9. To verify proper installation of software, verify that the T10 application folder is available from the program files of the PC.

*Note: In the event that a Windows Security warning dialogue box appears, select "Install this driver software anyway to continue the driver installation process" – This warning may appear multiple times during the installation.

*For the latest software version of iCAM T10 SDK, visit <http://www.irisid.com/ss/software>.

5.4 Installing the iCAM T10 USB Driver

The iCAM T10 has a USB driver that is supported with Windows XP and Windows 7 (x86 and x64 bit). The driver should automatically be installed when the software has been properly installed in the computer with compatible OS. Additionally, in the case of Windows 7 (x86 & x64), the drivers have been Windows (WHQL) certified and can be automatically detected using Windows 7 and an internet connection once the device has been connected to an available active USB 2.0 (or higher) port on the PC.

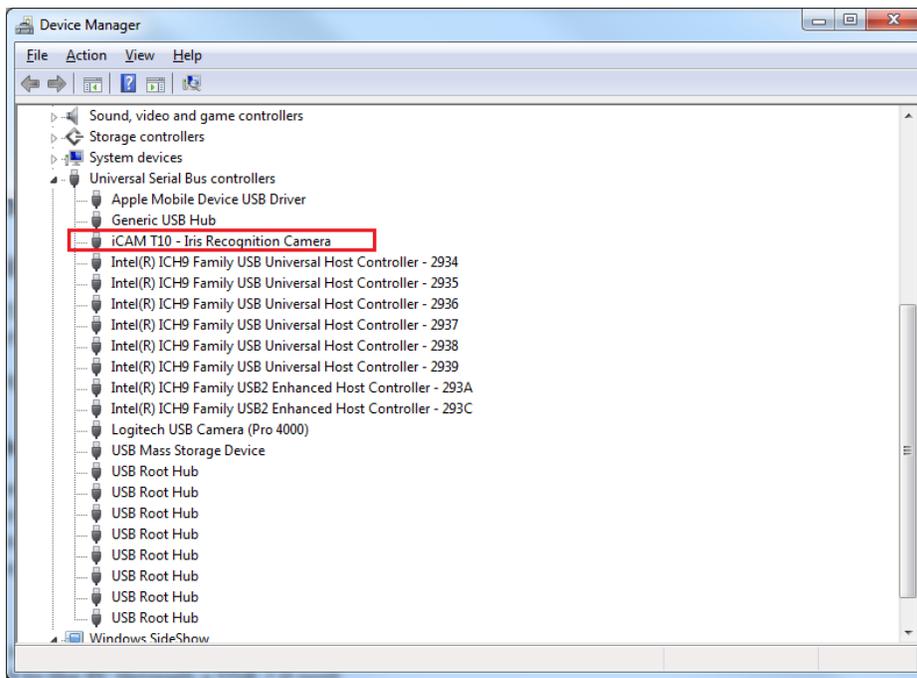
5.5 Verifying the iCAM T10 Device driver and manual driver installation

The proper installation of the driver will be automatically performed when a successful installation of the software is performed. However, in the event the driver is not properly installed, please see below for details on how to verify the driver installation, and how to perform a manual installation of the driver for T10.

**Note: To verify proper installation of drivers, open Device Manager - Universal Serial Controllers section (w/ unit connected).*

Follow the below steps to perform a verification test of the T10 driver:

1. Connect the iCAM T10 device to an available USB 2.0 port on the computer.
2. To verify proper installation of your device driver, view the contents of the Device Manager in Windows.
3. Locate the Universal Serial Controllers section of the device manager while the iCAM T10 is connected to the computer to determine if the driver is installed successfully as shown in the following image.



T10 device drivers can be installed in 3 different ways:

1. **Installing the T10 device driver using an available internet connection:**
 - T10 device Drivers have been Windows (WHQL) certified and can be automatically detected using Windows 7 and an internet connection once the device has been connected to an available active USB 2.0 (or higher) port on the PC.

**Note: In the event that this does not work appropriately, consider alternative driver installations listed in numbers 2, and 3.*

2. **Installing of the iCAM T10 device drivers using MSI installer package:**

- Using MSI installer package (iCAM_T10_Drivers_x86.msi or iCAM_T10_Drivers_x64.msi).

- These packages will create the following entries in 'Add/Remove Programs'.
 - 'iCAM T10 Drivers x86 v<driver_version>' on x86 platform
 - 'iCAM T10 Drivers x64 v<driver_version>' on x64 platform

**Note: This is a standalone installer and this does not require any other driver files for the installation.*

3. Installing the iCAM T10 device driver manually:

In the event that a manual installation of the device driver is required, follow the below steps:

To install the T10 driver manually, please follow the information found in this section of the document, making sure to follow the instructions for Windows 7 (x86) / Windows XP (x86) or Windows 7 (x64).

Windows XP and Windows 7 (x86) installation:

1. Connect the iCAM T10 to the USB2.0 port of the computer running Windows 7 or Windows XP 32-bit (x86). (If already connected, the "Found New Hardware Wizard" will automatically appear.)

**Note:* A message balloon "Found New Hardware" may appear. (This balloon message may close automatically as the "Found New Hardware Wizard" will appear).

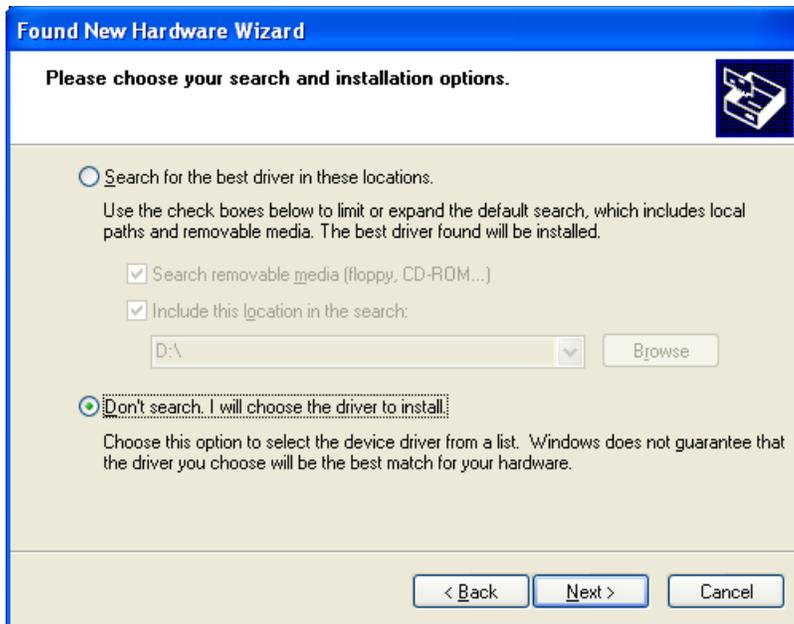
2. Select the radio button > **"Yes, now and every time I connect a device"** from the Found New Hardware Wizard Welcome screen and then press > **"Next"** as shown in the following image.



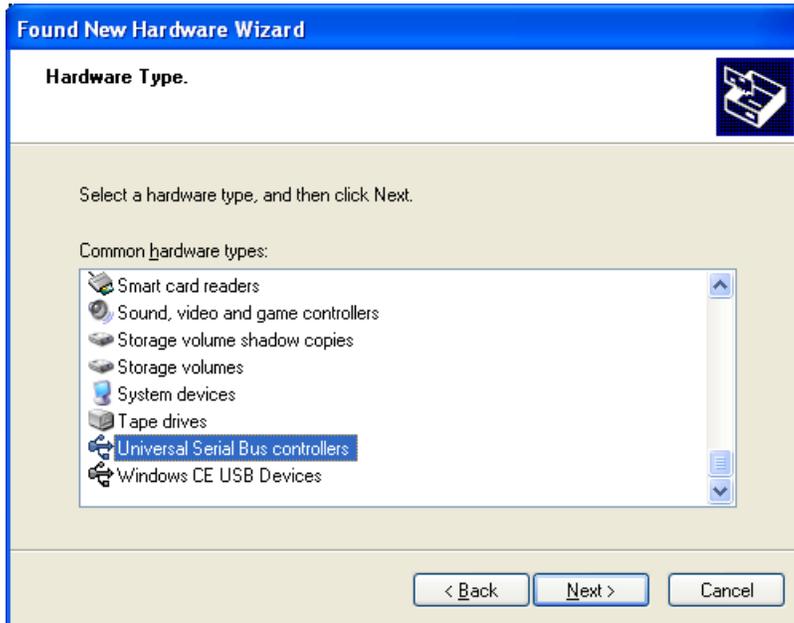
3. Select the radio button > "install from a specific location (advanced)" and then press > Next as shown in the following image.



4. Select > **“Don’t search. I will choose the driver to install.”** and press > **“Next”** as shown in the following image.



***Note:** If a “Hardware Type” screen comes up as shown in following image, select > **“Universal Serial Bus controllers”**. Then click > **“Next”**.

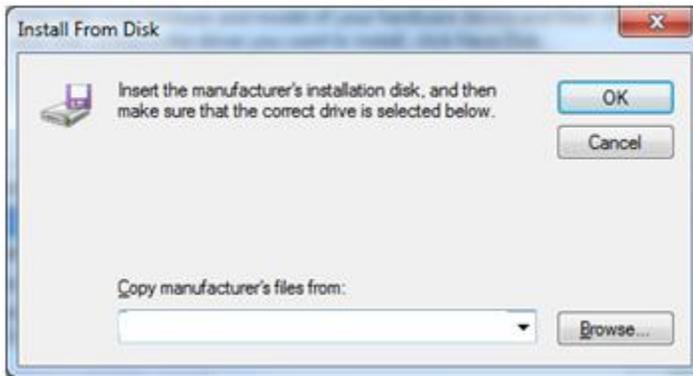


5. Press > **“Have Disk...”** as shown in following image.

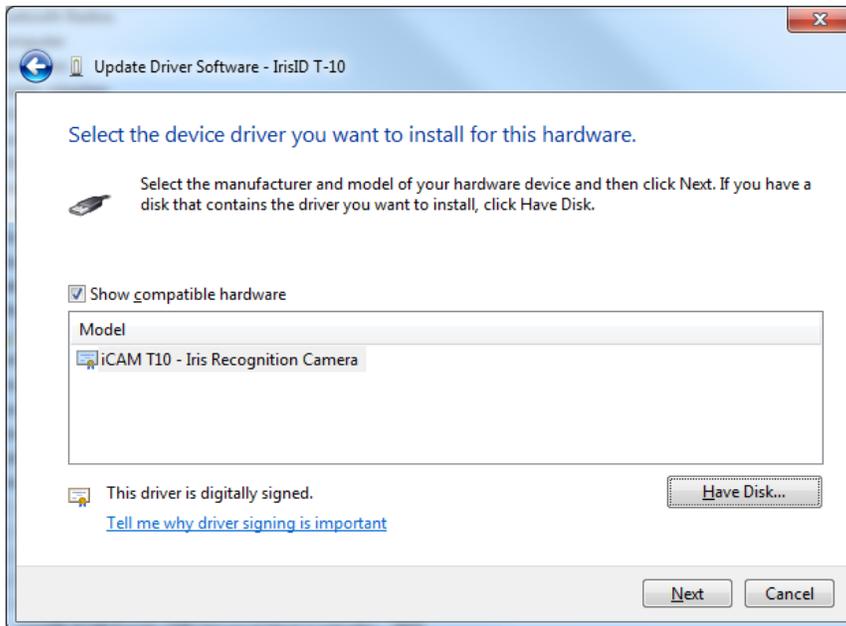


6. Press > **“Browse...”** as shown in following image.

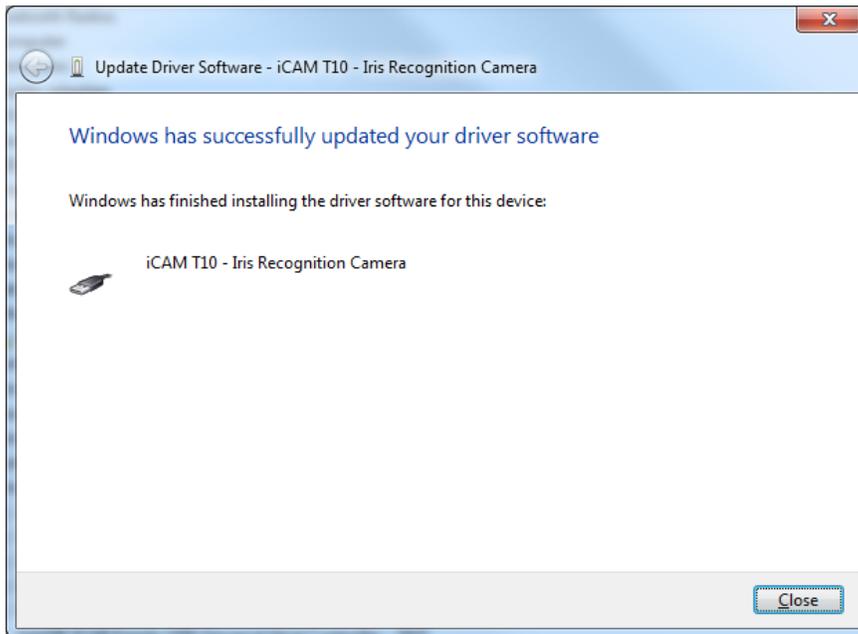
Browse for the Driver located on the iCAM T10 SDK Software build that has been provided. The drivers are located at: \T10 v.1.04.05\x86\T10SDK_SETUP_v1.04.05_x86\SETUP\program files\Iris ID\iCAM T10 SDK\Drivers\ [select windows ver. used] and press > Okay.



7. The “Select the device driver you want to install for this hardware: window may appear. Press > Next to continue as shown in the following image.



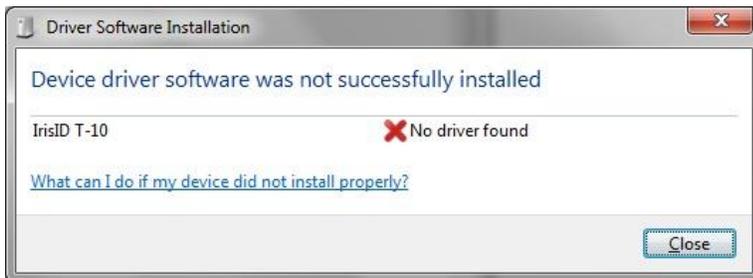
8. The driver will successfully install and a window prompt “Windows has successfully updated your driver” will be shown (as displayed in the following image).



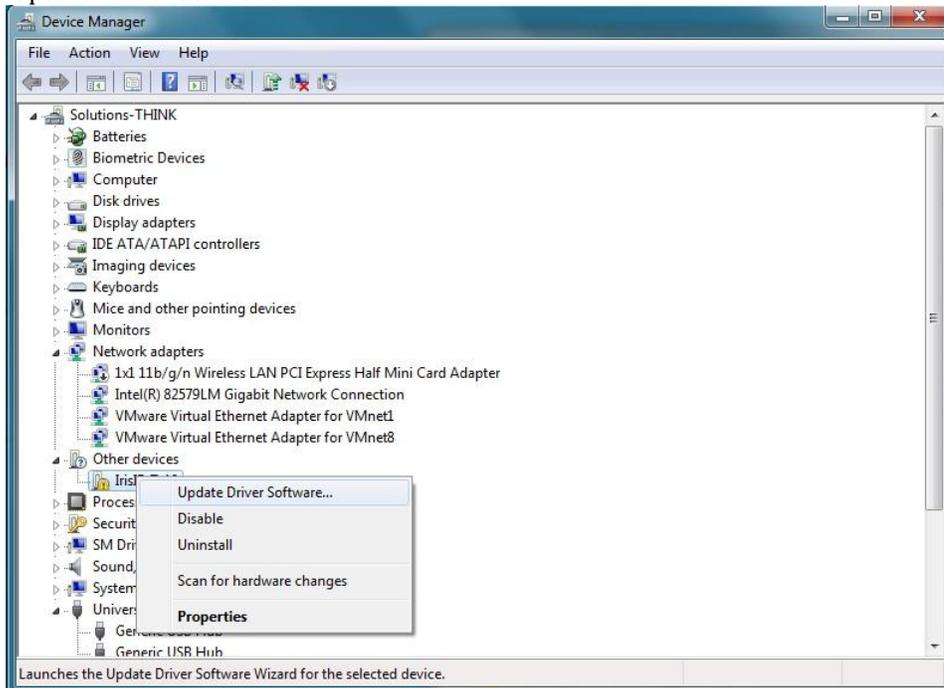
Windows 7 (x64) installation:

1. Connect the iCAM T10 to the USB2.0 port of the computer running Windows 7 64-bit (x64).

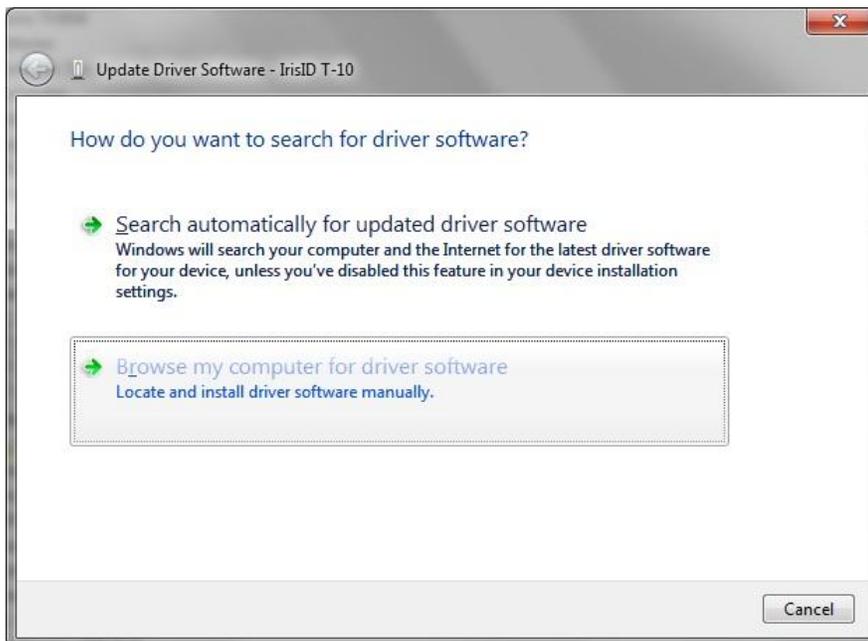
**Note:* A message window "Device driver software was not successfully installed" may appear. (Press "Close to close screen.)



2. Open the Device Manager within Windows and search for “Other devices – IrisID T-10” > right click “Update Driver Software...”.



3. Select > “Browse my computer for driver software – Locate and install driver software automatically”.

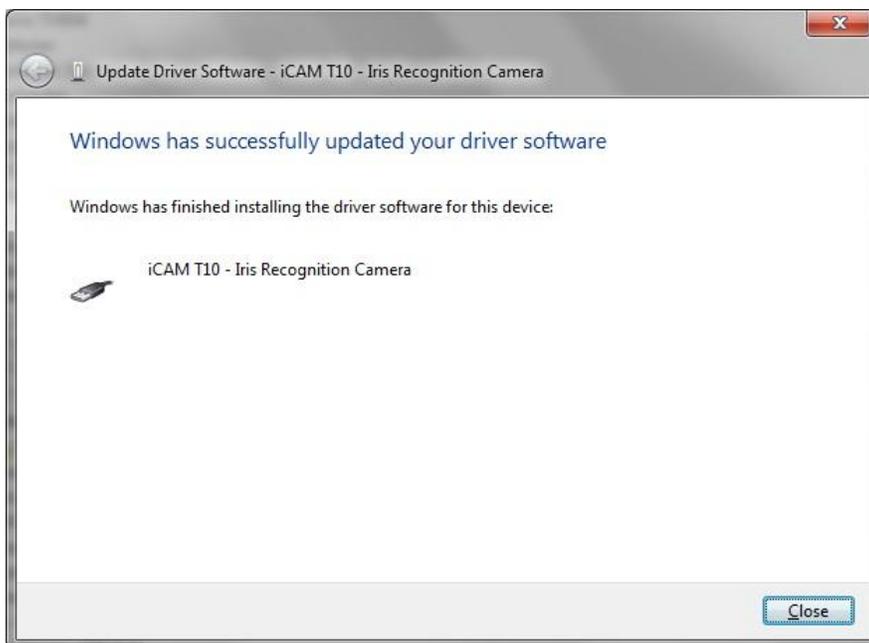


4. Browse for the driver software iCAM T10 Windows 7 (x64) at the following location: **C:\Program Files\Iris ID\Drivers\iCAM T10\T10DrvWin7** > and press **Next**.



**Note: install the iCAM T10 SDK software before attempting to browse to the driver location.*

5. Press > **“Close”** once the Window appears indicating that the driver was updated successfully.



5.6 Opening a Sample Application

1. Plug in the iCAM T10 device to the computer (must be plugged into a USB 2.0 port or above).
 - a. Ensure that the green power indicator LED is illuminated (to confirm power and connection).
 - b. Confirm that the iCAM device driver has been installed correctly.

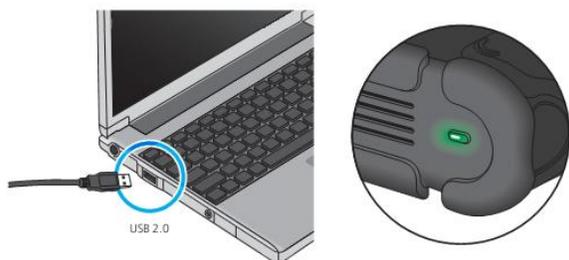


Fig. 1 T10 USB Cable connection and green light on device after connected

2. Press > **Connect** button (as shown in Fig. 1)

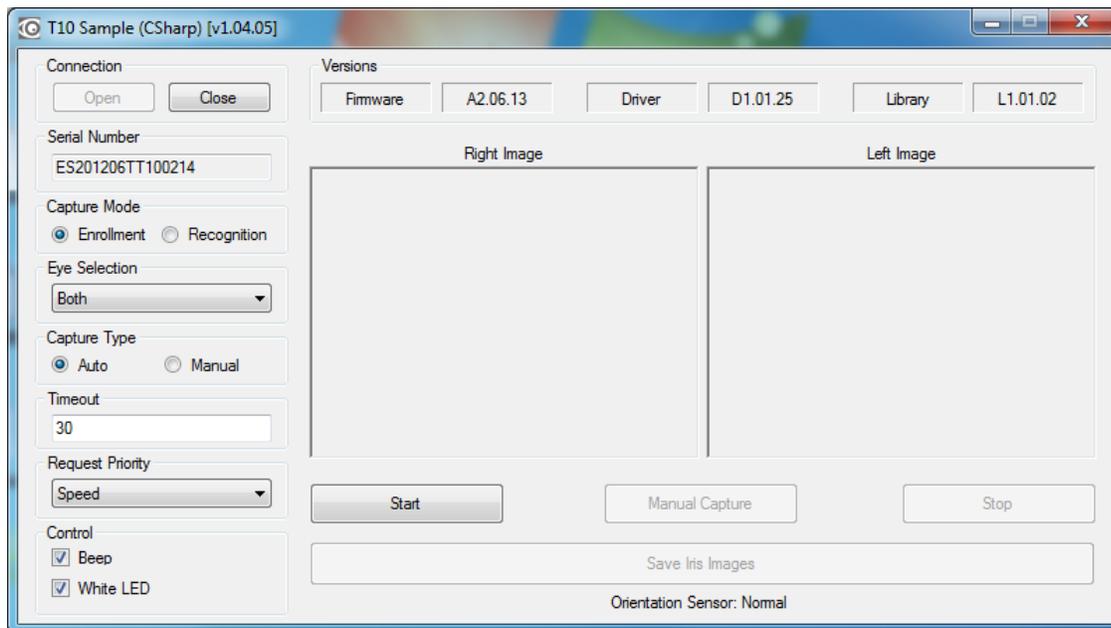


Fig. 2 Preview image of Sample SDK application

Upon successful connection, the sample application will appear as active with the version info of the SDK displayed (as shown in the following figure)

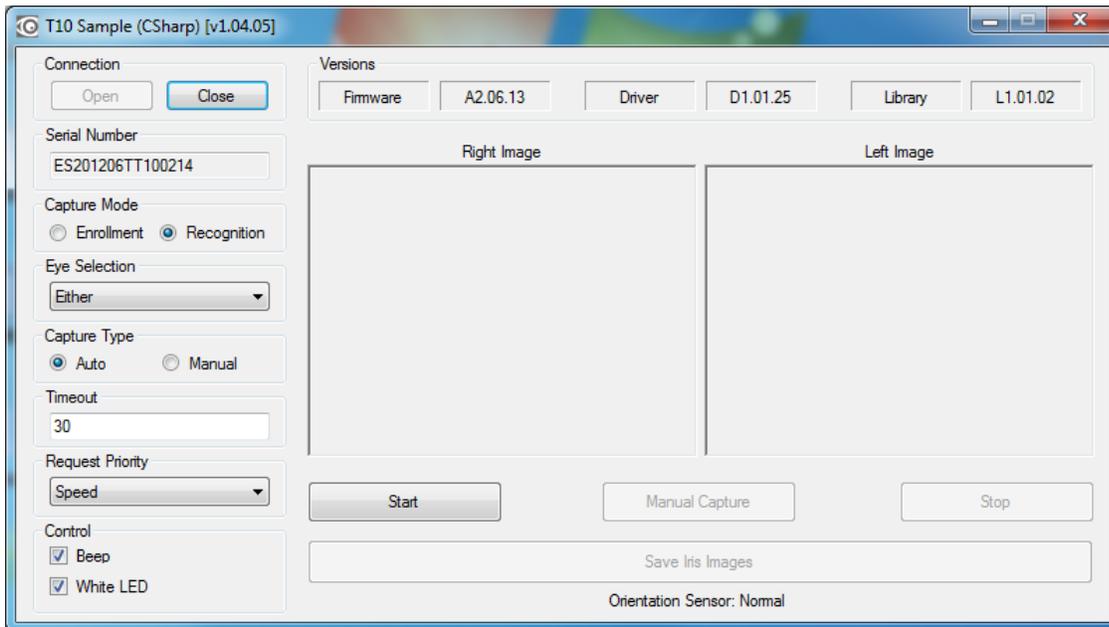


Fig. 3 SDK version info shown after the device connected.

5.7 Using the iCAM T10 Sample Application

Enrollment & Recognition

Use the “Eye Selection” dropdown box to specify how the iCAM will attempt to take an image. This can be specified for *enrollment* and *recognition* modes. The available options for selection are *Both*, *Left*, or *Right*. Additionally, when performing recognition, the option for “*Either eye*” is also available for use (using recognition mode selection only).

***Note:** *If the user has only one eye, select the eye selection for the eye that needs to be enrolled for automatic image to be taken of that user when using iris capture. For recognition in such instances, the setting of “Either Eye” or “Left”/“Right” eye can be used to properly capture an image automatically from the iCAM.*

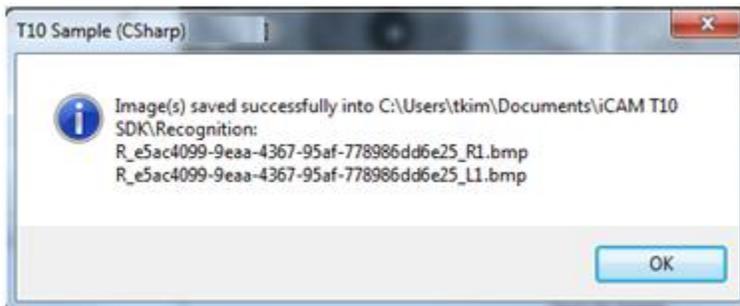
The enrollment and recognition functionality can be tested and images can be saved to view from **C:\users\[USER ID]\Documents\iCAM T10 SDK**.

***Note:** *The expression “Enroll User” refers to the exercise of capturing Iris images in an enrollment mode with our sample application(s). The image that is captured can be saved but will not be stored as a template for purposes of actual identification from the saved enrollment image(s). Template generation and matching are available in the iData Iris SDK.*

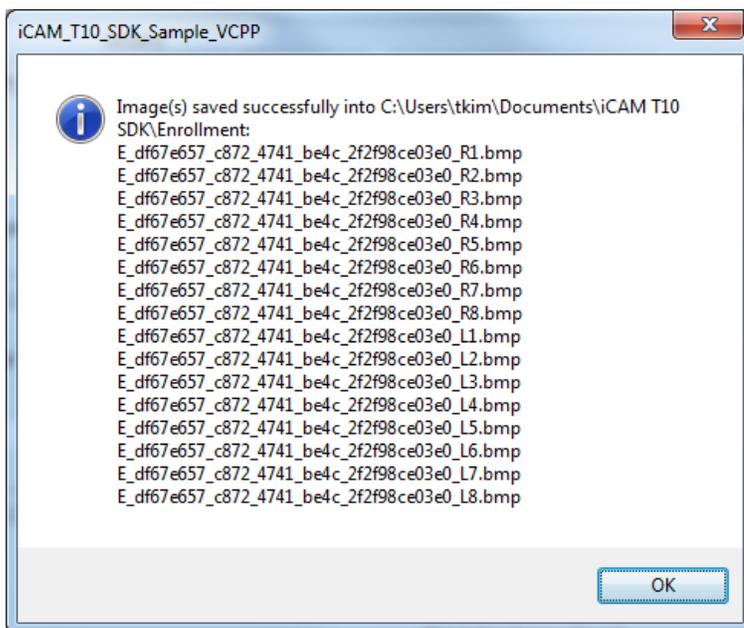
5.7.1 Steps to Capture Iris Enrollment Image (standard Steps)

1. Select Capture Mode> “Enrollment” radio button (if not already selected).
2. Select the “Eye Selection” type that will be used for enrollment. (If only one eye will be enrolled, select the correct eye to enroll (Left/Right)).

3. Select Capture Type “Auto” or “Manual”, which will be used for enrollment.
4. Type the timeout value from 0 to 30 (0 = continuous and 30 = maximum/default value in the sample application).
5. Select Request Priority “Speed”, “Quality”, or “Manual”.
6. Press > “Start” button to start iris capture process.
7. With the visor in position, look straight into the unit so that the left eye is centered in the mirror.
8. At this point in time the iCAM will automatically capture iris images (with Auto capture mode) that will be displayed on the “Iris Images” area of screen (If Manual capture mode is used, then the Manual Capture button must be pressed to capture).
9. Select the “Save Iris Images” button on the screen to save these images to the computer. A message box will be displayed indicating that the images have been saved. (The location of the saves files are: C:\users\[USER ID]\Documents\iCAM T10 SDK\Enrollment).



Iris Images – Save Iris Images with Speed (Request Priority)



Iris Images – Save Iris Images with Multiple (Request Priority)

**Note: The file name convention of sample application created is used for illustration purposes only.*

5.7.2 Steps to Recognize a User

**Note: The expression “Recognize a User” refers to the exercise of capturing Iris images in a recognition mode with our sample application(s). The image that is captured with this device control can be saved to disk, but will not be stored as a template for the purposes of actual identification from the saved enrollment/recognition image(s).*

1. Select Capture Mode > “Recognition” radio button (if not already selected).
2. Select the “which eye” type that will be used for recognition (Either eye mode can also be used).
3. Press > “Start” button to start iris capture process.
4. With the visor in position, look straight into the unit so that the left eye is centered in the mirror.
5. At this point in time the iCAM will automatically capture iris images (with Auto capture mode) that will be displayed on the “Iris Images” area of screen (If Manual capture mode is used, then the Manual Capture button must be pressed to capture).
6. Select the “Save Iris Images” button on the screen to save these images to the computer. A message box will be displayed indicating that the images have been saved. (The location of the saves files are: C:\users\[USER ID]\Documents\iCAM T10 SDK\Recognition).

5.8 How to Operate iCAM T10

Operating an iCAM T10 Camera unit is very intuitive. This section covers operation usage for general use, how to enroll a user, and recognition for an iCAM T10 camera unit.

5.8.1 Operation

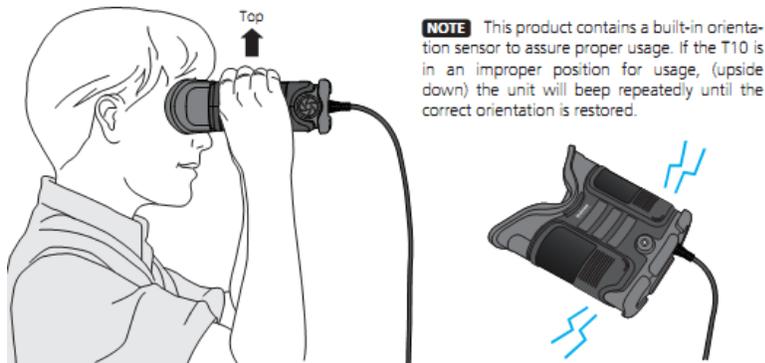
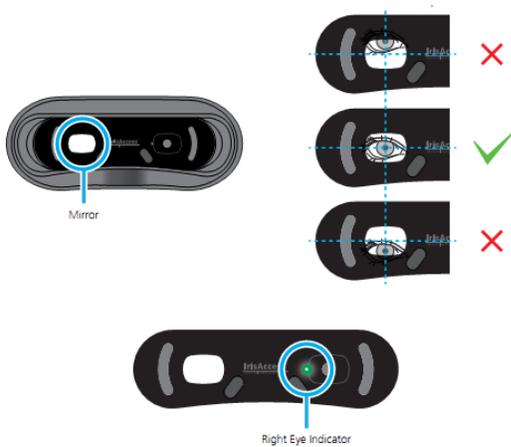


Fig. 4 Operation usage

- With the top of the unit facing up, gently hold the hand grips located at either side of the visor.
- Place the T10 in front of your eyes so that it is centered between the bridge of the nose and forehead (as shown in Figure 4).
- With the visor in position, look straight into the unit so that the left eye is centered in the mirror.
 - During the enrollment process the eyes should be open as wide as possible until the capture process is completed (as shown in figure 5).

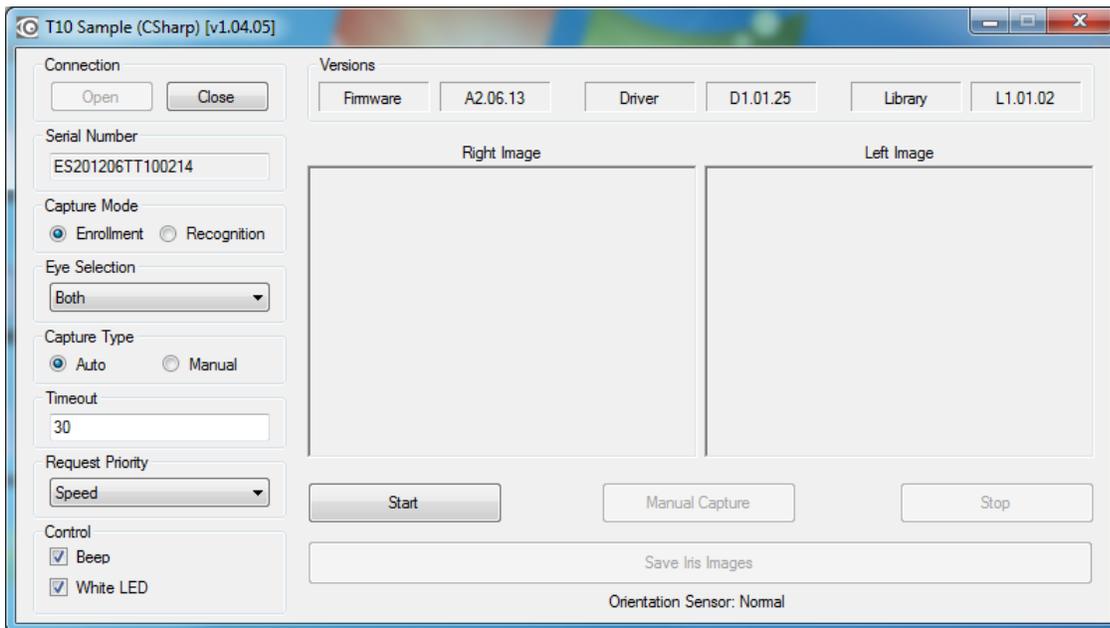
**Note: If looking into the mirror with the left eye is not possible or uncomfortable; the user may position the right eye straight at the right eye indicator LED. The unit operator may assist in directing the user during this process.*



Positioning eye(s) with visor

5.8.2 iCAM T10 SDK Sample Application Options & Features

Additional options for camera controls are available in the sample application. These settings allow for the following to be demonstrated:



iCAM T10 SDK sample application

1. **Connection** – Selectable Open and close options for connection/disconnection of T10 device.
 - a. **Open** – Connect the iCAM to the application
 - b. **Close** – Disconnects the iCAM from the application control.
2. **Serial Number** – Displays the serial number of the connected iCAM T10 device.
3. **Versions** – Displays the various versions for the following:
 - a. **Firmware** – Displays the firmware version of the connected iCAM T10 device.
 - b. **Driver** – Displays the driver version of the connected iCAM T10 device.
 - c. **Library** – Displays the library version of the connected iCAM T10 device.

4. **Capture Mode** - Part of the iris Camera section of the sample SDK, this specifies the represented type of image that is to be taken. Capture options available in the sample application(selectable by radio button) are:
 - a. Enrollment
 - b. Recognition
5. **Eye Selection** - Determines the eye that is to be used for the image capture. Select the type that will be used for enrollment or for recognition capture. (If only one eye will be enrolled, select the correct eye to recognize (Left/Right). Either eye mode can also be used if needed when set to recognition mode only).
6. **Capture Type** - Selectable by radio button, this option allows the user to select either "Auto" or "Manual" for image capture.
 - a. **Auto** - Automatically capture image(s) when the subject is in range.
 - b. **Manual** - Allows the user to manually capture an image.
7. **Timeout** - Iris Camera timeout for the capture event initiated. This timing setting is represented in seconds. (Timeout values range from 0-30 seconds with 0 being infinite and 30 being maximum seconds that can be specified.)
8. **Request Priority** - The following three Modes are available only Automatic Capture Mode. In the Manual Capture Mode only Speed mode is available.
 - a. **Speed** - The first qualified (as a valid iris image) is transmitted. (Available in "Manual Capture Mode" ONLY.)
 - b. **Quality** - The best qualified (with the highest quality) out of latest eight qualified images is transmitted.
 - c. **Multiple** - All the eight qualified iris images are transmitted. 3rd party quality measurement algorithm may be used to pick a 'best' image.
9. **Control**
 - a. **Beep** - Enable/Disable selectable (When enabled, the T10 beeper will sound continuously when the device is tilted upside down (and the "start" Button has been pressed). Additionally, the beeper will beep after successful capture of iris images).
 - b. **White LED** - Enable/Disable selectable (When enabled, white LED is on).
10. **Iris Images** - Displays the right and/or left captured iris image(s).
11. **Start** - Initiates the T10 device to an active state for acquiring an iris image.
12. **Manual Capture** - When active, allows the user to manually press the button to force a capture of the image provided by the T10 device.
13. **Stop** - halts and disengages the active state of the T10.
14. **Save Iris Images** - Captured Enrollment and / or recognition images can be saved to the computer. A message box will display indicating a successful save and the location of the saved images. (The location of the saves files for Enrollment & Recognition are:
 - a. **Enrollment Images:** C:\users\[USER ID]\Documents\iCAM T10 SDK\Enrollment
 - b. **Recognition Images:** C:\users\[USER ID]\Documents\iCAM T10 SDK\Recognition
15. **Orientation Sensor** - Displays the active state of the T10 devices physical state. When correctly positioned, the "Normal" text will appear. If upside-down, the camera will not be able to take an iris image (and may beep if the "Beep" control is selected). Additionally, when the orientation is incorrect, the message will display in red as "upside down".

6. iCAM T10 SDK Device Control

6.1 Overview

Environment

- Operating System: Windows XP and Windows 7.
- Support: Visual Studio 2008 SP1 (and above).
- Platform: Pentium compatible PC with USB port
- Included dlls:

- iCamT10Control.dll (C++ SDK component for device management)
- iCAMT10ControlAx.dll (COM SDK component for device management)
- libT10EyeSeek.dll (Iris image processing library)
- Interop.iCAMT10ControlAxLib.dll (Interop dll)

- Included T10SDK.h (Header file for T10 C++ SDK)

- Included lib:

- iCamT10Control.lib (Import library for iCamT10Control dll)
- iCAMT10ControlAx.tlb (Type library for device SDK COM component)

Purpose

This SDK module provides APIs that control an iCAM T10 camera to capture 640x480 iris images for enrollment and recognition.

6.2 Software Development Compatibility

The iCAM T10 sample applications were developed using VS2008 SP1.

NOTE: The use of a different version of Visual Studio may require changes when trying to compile the application in that version of Visual studio.

6.3 Software Architecture

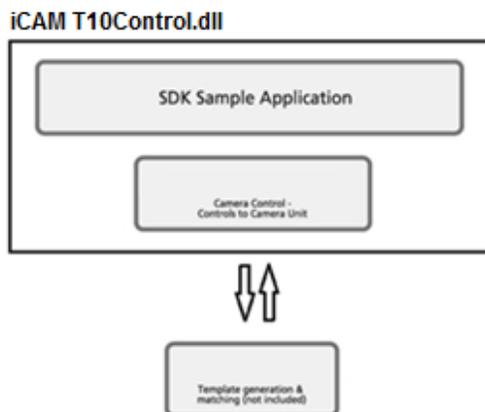


Fig. 10 Software architecture of iris recognition using iCAM T10 SDK

The iCAM needs the ICAMT10Control.dll (iCAMT10ControlAx.dll) and iDataIris.dll for a complete implementation of a host based iris recognition application.

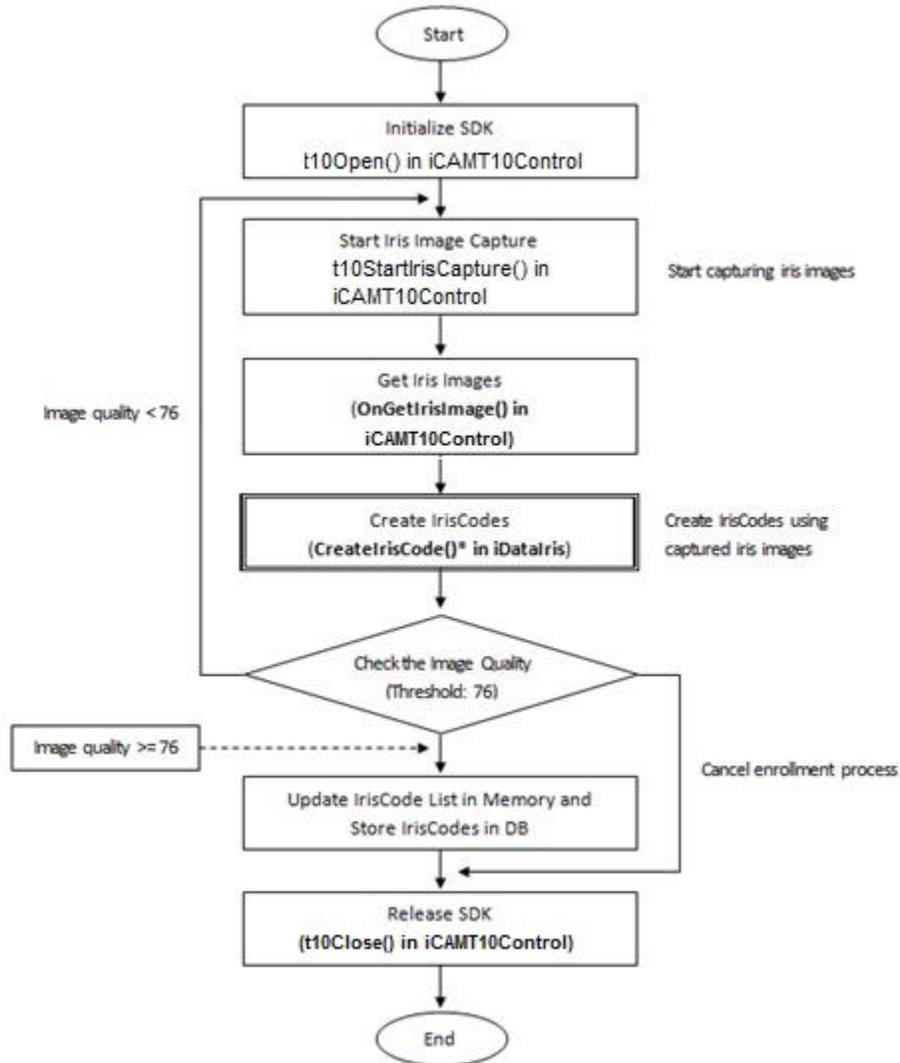
- ICAMT10Control.dll will get standards compliant iris images from an iCAM T10 series camera.
- iDataIris.dll is used for creating and verifying iris templates and small scale local host matching functions when writing your application. (This is not required for standard usage of the sample applications provided.)
- For large scale 1:N matching applications, the IrisAccelerator™ is used see: <http://www.irisid.com/irisaccelerator>

**Note:* The entry level iCAM T10 SDK does not include iDataIris.dll (matching and quality assessment functions). If matching and quality assessment is needed, please contact Iris ID Systems, Inc. at: sales@irisid.com or call +1-609-819-4747 and select option 2.

6.4 General Procedures for Capturing Images

Enrollment Procedure:

The below diagram is an overview of the general procedure for capturing images as enrollment. The diagram steps through the capture process from start to end mapped out in a logical view (for reference purposes only).

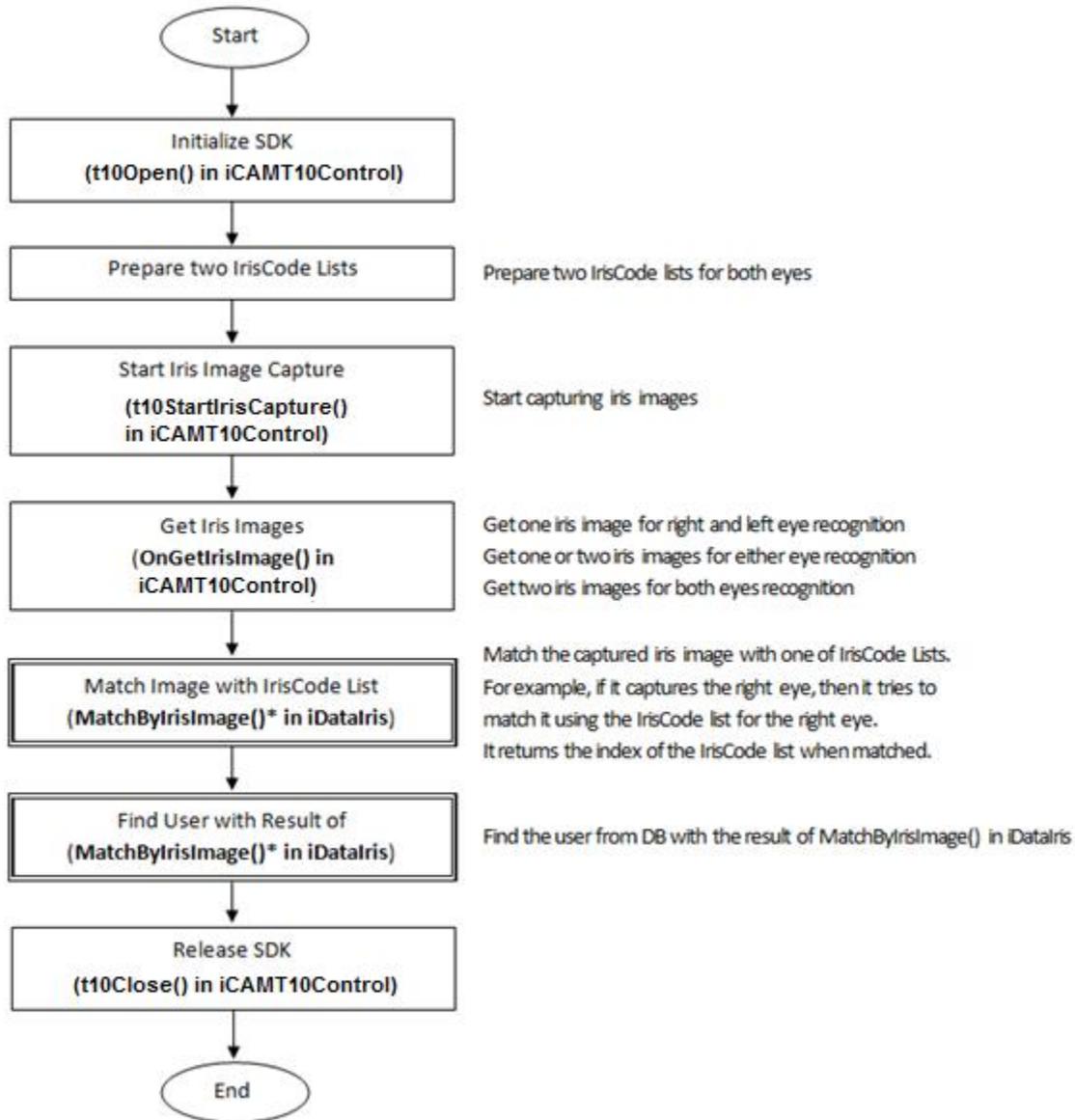


**iCAMT10 SDK does not include "CreateIrisCode" API. Please use iData Iris SDK to generate iris templates.*

****NOTE: it is strongly recommended to enroll both eyes of each user for the best performance.**

Identification Procedure:

The below diagram is an overview of the general procedure for capturing images as identification. The diagram steps through the capture process from start to end mapped out in a logical view (for reference purposes only).



**iCAMT10 SDK does not include "MatchByIrisImage" API. Please use iData Iris SDK to match.*

6.5 Application Programming Interface Function & Event Summary

Programming development can be performed using either C/C++ or C#.

Please view appropriate function listing for development method that is to be used.

Application Programming Interface Functions

Note: **t10 prefix of each API name is for C/C++ development only. **t10** prefix is not used in C# development. For example, **t10GetVersion** API in C/C++ is expressed as **GetVersion** API in C#.*

C/C++

t10GetVersion()	To get the version information of SDK, firmware, driver, and library.
t10Open()	To initialize the SDK and connect to the iCAM.
t10Close()	To close the SDK and disconnect from the iCAM.
t10GetSerialNumber()	To get the serial number of the connected iCAM.
t10SetMoveIrisPosition()	To move iris position.
t10SetBeeper()	To set the beeper as enabled or disabled.
t10SetIllumination()	To set the white illumination as enabled or disabled.
t10StartIrisCapture()	To start capturing iris images from the iCAM.
t10CaptureIris()	To get iris images from the iCAM for enrollment or recognition.
t10AbortIrisCapture()	To cancel iris capture operation.
t10SetCallback()	To wire up event callbacks.

C#

GetVersion()	To get the version information of SDK, firmware, driver, and library.
Open()	To initialize the SDK and connect to the iCAM.
Close()	To close the SDK and disconnect from the iCAM.
GetSerialNumber()	To get the serial number of the connected iCAM.
SetMoveIrisPosition()	To move iris position.
SetBeeper()	To set the beeper as enabled or disabled.
SetIllumination()	To set the white illumination as enabled or disabled.
StartIrisCapture()	To start capturing iris images from the iCAM.
CaptureIris()	To get iris images from the iCAM for enrollment or recognition.
AbortIrisCapture()	To cancel iris capture operation.

Application Event Functions

Application event functions are functions used in an application that define or determine an event to occur. The items in this table are results from another action.

Note: **Fire_On prefix is for C/C++ development only. **Fire_On** prefix is not used in C# development. For example, **Fire_OnGetStatus** in C/C++ is expressed as **GetStatus** in C#.*

C/C++

Fire OnGetStatus	To get the status of the iCAM.
Fire ONGetLiveInfo	To get the live iris images after calling StartIrisCapture().
Fire OnGetIrisImage	To get the captured iris images.

C#

GetStatus	To get the status of the iCAM.
GetLiveInfo	To get the live iris images after calling StartIrisCapture().
GetIrisImage	To get the captured iris images.

**Note: Some functions are non-blocking and other s are blocking.*

6.6 Application Programming Interface

6.6.1 t10GetVersion()

C/C++

t10GetVersion()			
Syntax	t10GetVersion (LONG lType, CHAR* szVersion)		
	Argument	Type	Description
	lType	[in] LONG	The type of version required. <ul style="list-style-type: none"> - SDK_VER_SDK - SDK_VER_FIRMWARE - SDK_VER_DRIVER - SDK_VER_LIBRARY
	szVersion	[out] CHAR*	Version information of required type.
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	It gets the SDK, firmware, driver, and library versions.		
Notes	SDK version can be retrieved any time but for rest, the SDK needs to be opened first.		

C#

GetVersion()			
Syntax	GetVersion (LONG lType, BSTR* pbstrVersion)		
	Argument	Type	Description
	lType	[in] LONG	The type of version required. <ul style="list-style-type: none"> - SDK_VER_SDK - SDK_VER_FIRMWARE - SDK_VER_DRIVER - SDK_VER_LIBRARY
	pbstrVersion	[out] BSTR*	Version information of required type.
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	It gets the SDK, firmware, driver, and library versions.		
Notes	SDK version can be retrieved any time but for rest, the SDK needs to be opened first.		

6.6.2 t10Open()

C/C++

t10Open()			
Syntax	t10Open()		
	Argument	Type	Description
	NONE		
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	Opens connection to the SDK.		
Notes	The beeper and white LED illumination are enabled during t10Open() call by default. The callback events should be wired up by the client application before making the t10Open() call.		

C#

Open()			
Syntax	Open()		
	Argument	Type	Description
	NONE		
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	Opens connection to the SDK.		
Notes	The beeper and white LED illumination is enabled during Open() call by default. The callback events should be wired up by the client application before making the Open() call.		

6.6.3 t10Close()

C/C++

t10Close()			
Syntax	t10Close()		
	Argument	Type	Description
	NONE		
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	Closes connection to the SDK.		

C#

Close()			
Syntax	Close()		
	Argument	Type	Description
	NONE		
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	Closes connection to the SDK.		

6.6.4 t10GetSerialNumber()

C/C++

t10GetSerialNumber()			
Syntax	t10GetSerialNumber(CHAR* pszSerialNumber)		
	Argument	Type	Description
	pszSerialNumber	[out] CHAR*	Serial number.
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	Gets the serial number of the connected T10 device.		
	The SDK should be opened before executing this call. The user has to allocate buffer for receiving the serial number data.		

C#

GetSerialNumber()			
Syntax	GetSerialNumber(BSTR* pbstrSerialNumber)		
	Argument	Type	Description
	pbstrSerialNumber	[out] BSTR*	Serial number.
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	Gets the serial number of the connected T10 device.		
	The SDK should be opened before executing this call. The user has to allocate buffer for receiving the serial number data.		

6.6.5 t10SetMoveIrisPosition()

C/C++

t10SetMoveIrisPosition()			
Syntax	t10SetMoveIrisPosition(LONG lEye, LONG lXPos, LONG lYPos)		
	Argument	Type	Description
	lEye	[in] LONG	Selected eyes. - SDK_EYE_RIGHT - SDK_EYE_LEFT
	lXPos	[in] LONG	X-axis position 0~639
	lYPos	[in] LONG	Y-axis position 0~479
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	<p>In the Manual Capture Mode, when an eye image is not fully viewable inside the 640x480 window, operator can translate/shift the 640x480 window so as to include the eye image at the center, in order to capture the optimal iris image.</p>		
Note	This feature is not available in Automatic Capture Mode.		

C#

SetMoveIrisPosition()			
Syntax	SetMoveIrisPosition(LONG lEye, LONG lXPos, LONG lYPos)		
	Argument	Type	Description
	lEye	[in] LONG	Selected eyes. - SDK_EYE_RIGHT - SDK_EYE_LEFT

	IXPos	[in] LONG	X-axis position 0~639
	IYPos	[in] LONG	Y-axis position 0~479
Return Type	LONG	0 for success. Please refer appendix for error code description.	
Purpose	<p>In the Manual Capture Mode, when an eye image is not fully viewable inside the 640x480 window, operator can translate/shift the 640x480 window so as to include the eye image at the center, in order to capture the optimal iris image.</p>		
Note	This feature is not available in Automatic Capture Mode.		

6.6.6 t10SetBeeper()

C/C++

t10SetBeeper()			
Syntax	t10SetBeeper(LONG lEnable)		
	Argument	Type	Description
	lEnable	[in] long	Enable status <ul style="list-style-type: none"> - DISABLED - ENABLED
Return Type	long	0 for success. Please refer to 'Error Codes' section for more information.	
Purpose	Enables or disables the beeper.		
Notes	If the beeper is enabled: <ul style="list-style-type: none"> - The beeper will sound continuously when the device is tilted upside down. - The beeper will beep after successful capture of iris images. 		

C#

SetBeeper()			
Syntax	SetBeeper(LONG lEnable);		
	Argument	Type	Description
	lEnable	[in] LONG	Enable status <ul style="list-style-type: none"> - DISABLED - ENABLED
Return Type	LONG	0 for success. Please refer to 'Error Codes' section for more information.	
Purpose	Enables or disables beeper.		
Notes	If Beeper is enabled: <ul style="list-style-type: none"> - Beeper will sound continuously when the device is tilted upside down. - The beeper will beep after successful capture of iris images. 		

6.6.7 t10SetIllumination()

C/C++

t10SetIllumination()			
Syntax	t10SetIllumination(long lEnable)		
	Argument	Type	Description
	lEnable	[in] long	Enable status - DISABLED - ENABLED
Return Type	long	0 for success. Please refer to 'Error Codes' section for more information.	
Purpose	It controls white illumination for the contraction of the pupil.		

C#

SetIllumination()			
Syntax	SetIllumination(LONG lEnable);		
	Argument	Type	Description
	lEnable	[in] LONG	Enable status - DISABLED - ENABLED
Return Type	LONG	0 for success. Please refer to 'Error Codes' section for more information.	
Purpose	It controls white illumination for the contraction of the pupil.		

6.6.8 t10StartIrisCapture()

C/C++

t10StartIrisCapture()			
Syntax	t10StartIrisCapture(long lMode, long lType, long lEye, long lPriority, long lTimeout)		
	Argument	Type	Description
	lMode	[in] long	Capture modes. <ul style="list-style-type: none"> - SDK_MODE_RECOGNITION - SDK_MODE_ENROLLMENT
	lType	[in] long	Capture types. <ul style="list-style-type: none"> - SDK_CAPTURE_TYPE_AUTO - SDK_CAPTURE_TYPE_MANUAL
	lEye	[in] long	Selected eye(s) to be captured. <ul style="list-style-type: none"> - SDK_EYE_RIGHT - SDK_EYE_LEFT - SDK_EYE_BOTH - SDK_EYE_EITHER SDK_EYE_EITHER option for is NOT valid for <i>lMode</i> == SDK_MODE_ENROLLMENT
	lPriority	[in] long	Priority mode <ul style="list-style-type: none"> - SDK_PRIORITY_SPEED - SDK_PRIORITY_QUALITY - SDK_PRIORITY_MULTIPLE All these three Modes are available only Automatic Capture Mode. In the Manual Capture Mode only Speed mode is available. SPEED MODE: the first qualified (as a valid iris image) is transmitted. QUALITY MODE: the best qualified (with the highest quality) out of latest eight qualified images is transmitted. MULTIPLE MODE: all the eight qualified iris images are transmitted. 3rd party quality measurement algorithm may be used to pick a 'best' image.
lTimeout	[in] long	Timeout for capture operation. Parameter range: 0~30 The capture operation will NOT timeout if <i>lTimeout</i> == 0.	
Return Type	long	0 for success. Please refer to 'Error Codes' section for more information.	

Purpose	Starts iris capture operation.
---------	--------------------------------

C#

StartIrisCapture()			
Syntax	StartIrisCapture(LONG lMode, LONG lType, LONG lEye, LONG lPriority, LONG lTimeout);		
	Argument	Type	Description
	lMode	[in] LONG	Capture modes. <ul style="list-style-type: none"> - SDK_MODE_ENROLLMENT - SDK_MODE_RECOGNITION
	lType	[in] LONG	Capture types. <ul style="list-style-type: none"> - SDK_CAPTURE_TYPE_AUTO - SDK_CAPTURE_TYPE_MANUAL
	lEye	[in] LONG	Selected eye(s) to be captured. <ul style="list-style-type: none"> - SDK_EYE_RIGHT - SDK_EYE_LEFT - SDK_EYE_BOTH - SDK_EYE_EITHER <p>SDK_EYE_EITHER option for is NOT valid for <i>lMode</i> == SDK_MODE_ENROLLMENT</p>
	lPriority	[in] LONG	Priority mode <ul style="list-style-type: none"> - SDK_PRIORITY_SPEED - SDK_PRIORITY_QUALITY - SDK_PRIORITY_MULTIPLE <p>All these three Modes are available only Automatic Capture Mode. In the Manual Capture Mode only Speed mode is available.</p> <p>SPEED MODE: the first qualified (as a valid iris image) is transmitted.</p> <p>QUALITY MODE: the best qualified (with the highest quality) out of latest eight qualified images is transmitted.</p> <p>MULTIPLE MODE: all the eight qualified iris images are transmitted. 3rd party quality measurement algorithm may be used to pick a 'best' image.</p>
lTimeout	[in] LONG	Timeout for capture operation. Parameter range: 0~30 The capture operation will NOT timeout if <i>lTimeout</i> == 0.	

Return Type	LONG	0 for success. Please refer to 'Error Codes' section for more information.
Purpose	Starts iris capture operation.	

6.6.9 t10CaptureIris()

C/C++

t10CaptureIris()		
Syntax	t10CaptureIris()	
	Argument	Type
	NONE	
Return Type	long	0 for success. Please refer to 'Error Codes' section for more information.
Purpose	Iris capture is triggered by the host PC using this command. This is available in SDK_CAPTURE_TYPE_MANUAL only.	
Details	In case the capture type is "Auto", then the priority is automatically changed to "SDK_PRIORITY_SPEED" during software triggered capture.	

C#

CaptureIris();		
Syntax	CaptureIris();	
	Argument	Type
	NONE	
Return Type	LONG	0 for success. Please refer to 'Error Codes' section for more information.
Purpose	Iris capture is triggered by the host PC using this command. This is available in SDK_CAPTURE_TYPE_MANUAL only.	
Details	In case the capture type is "Auto", then the mode is automatically changed to "SDK_CAPTURE_TYPE_MANUAL" and priority to "SDK_PRIORITY_SPEED" during software triggered capture.	

6.6.10 t10AbortIrisCapture()

C/C++

t10AbortIrisCapture()			
Syntax	t10AbortIrisCapture()		
	Argument	Type	Description
	NONE		
Return Type	long	0 for success. Please refer to 'Error Codes' section for more information.	
Purpose	Cancels the ongoing capturing operation.		
Details	The operation is valid only when capture is in progress.		

C#

AbortIrisCapture();			
Syntax	AbortIrisCapture();		
	Argument	Type	Description
Return Type	LONG	0 for success. Please refer to 'Error Codes' section for more information.	
Purpose	Cancels the ongoing capturing operation.		
Details	The operation is valid only when capture is in progress.		

6.6.11 T10SetCallback()

C/C++

t10SetCallback()			
Syntax	t10SetCallback(long lEventType, void* pfn, void * pvContext);		
	Argument	Type	Description
	lEventType	[in] long	Type of event function pointer. <ul style="list-style-type: none"> - SDK_CB_GET_IRIS_IMAGE - SDK_CB_GET_LIVE_INFO - SDK_CB_GET_STATUS
	pfn	[in] void*	Function pointer as per <i>lEventType</i> .
	pvContext	[in] void*	User provided context that is returned by SDK during callback.
Return Type	long	0 for success. Please refer to 'Error Codes' section for more information.	
Purpose	Wires up event callback mechanism.		
Details	The events should be wired up before "t100open()" call, else the application may fail to get some initial set of events. Events can only be wired when the device is not capturing.		
Notes	Applies to C SDK only.		

C#: SetCallBack API is not applicable

6.7 Events

6.7.1 Fire_OnGetStatus

C/C++

Fire_OnGetStatus			
Syntax	<code>typedef void (*Fire_OnGetStatus)(long IType, long IValue, void * pContext)</code>		
	Argument	Type	Description
	IType	[in] long	Status type <ul style="list-style-type: none"> - SDK_EVENT_DEVICE_TILT - SDK_EVENT_SURPRISE_REMOVE - SDK_EVENT_TXN_TIMEOUT - SDK_EVENT_TXN_ERROR
	IValue	[in] long	Data value of event
	pContext	[in] void*	User provided value in "t10SetCallback()" is returned by the event.
Purpose	It notifies that the status of device.		

C#

GetStatus			
Syntax	<code>GetStatus(LONG IType, LONG IValue);</code>		
	Argument	Type	Description
	IType	[in] LONG	Status type <ul style="list-style-type: none"> - SDK_EVENT_DEVICE_TILT - SDK_EVENT_SURPRISE_REMOVE - SDK_EVENT_TXN_TIMEOUT - SDK_EVENT_TXN_ERROR
	IValue	[in] LONG	Data value of event
Purpose	It notifies that the status of device.		

6.7.2 Fire_OnGetLiveInfo

C/C++

Fire_OnGetLiveInfo			
Syntax	<pre>typedef void (*Fire_OnGetLiveInfo)(long lEye,long lWidth,long lHeight,unsigned char * pucLiveImage,long lImageSize, void * pContext)</pre>		
	Argument	Type	Description
	lEye	[in] long	Selected eye <ul style="list-style-type: none"> - SDK_EYE_RIGHT - SDK_EYE_LEFT - SDK_EYE_BOTH - SDK_EYE_EITHER
	lWidth	[in] long	Width of live image
	lHeight	[in] long	Height of live image
	pucLiveImage	[in] unsigned char*	Live image
	lImageSize	[in] long	Size of live image
	pContext	[in] void*	User provided value in "t10SetCallback()" is returned by the event.
Purpose	Callback for live image information.		

C#

GetLiveInfo			
Syntax	<pre>GetLiveInfo(LONG lEye, LONG lWidth, LONG lHeight, VARIANT vLiveImage, LONG lImageSize);</pre>		
	Argument	Type	Description
	lEye	[in] LONG	Selected eye <ul style="list-style-type: none"> - SDK_EYE_RIGHT - SDK_EYE_LEFT - SDK_EYE_BOTH - SDK_EYE_EITHER
	lWidth	[in] LONG	Width of live image
	lHeight	[in] LONG	Height of live image
	vLiveImage	[in] VARIANT	Live image
	lImageSize	[in] LONG	Size of live image
Purpose	Callback for live image information.		

6.7.3 Fire_OnGetIrisImage

C/C++

Fire_OnGetIrisImage			
Syntax	<pre>typedef void (*Fire_OnGetIrisImage)(long lEye, long lNumFrameToReturn, long lWidth, long lHeight, unsigned char *puIrisImage, long lImageSize, void * pContext)</pre>		
	Argument	Type	Description
	lEye	[in] long	Selected eye <ul style="list-style-type: none"> - SDK_EYE_RIGHT - SDK_EYE_LEFT - SDK_EYE_BOTH - SDK_EYE_EITHER
	lNumFrameToReturn	[in] long	The frame number can vary from 1 ~ 8 depending on capture priority. In case of "MULTIPLE", 8 frames of each eye is returned and for other options, only one frame per eye is returned.
	lWidth	[in] long	Width of captured iris image
	lHeight	[in] long	Height of captured iris image
	puIrisImage	[in] unsigned char*	Iris image
	lImageSize	[in] long	Size of iris image.
	pContext	[in] void*	User provided value in "t10SetCallback()" is returned by the event.
Purpose	Callback for iris image information.		
Notes	This event fires once for each eye. In case of priority mode = MULTIPLE, this event will fire 8 times for each eye.		

C#

GetIrisImage			
Syntax	<pre>GetIrisImage(LONG lEye, LONG lFrameNumber, LONG lWidth, LONG lHeight, VARIANT vIrisImage, LONG lImageSize);</pre>		
	Argument	Type	Description
	lEye	[in] LONG	Selected eye <ul style="list-style-type: none"> - SDK_EYE_RIGHT - SDK_EYE_LEFT - SDK_EYE_BOTH - SDK_EYE_EITHER
	lFrameNumber	[in] LONG	The frame number can vary from 1 ~ 8 depending on

			priority. In case of "MULTIPLE", 8 frames of each eye is returned and for other options, only one frame per eye is returned.
	IWidth	[in] LONG	Width of captured iris image
	IHeight	[in] LONG	Height of captured iris image
	vIrisImage	[in] VARIANT	Iris image
	IImageSize	[in] LONG	Size of iris image.
Purpose	Callback for iris image information.		
Notes	This event fires once for each eye. In case of priority mode = MULTIPLE, this event will fire 8 times for each eye.		

6.8 Constants

Version types	
0	SDK_VER_SDK
1	SDK_VER_FIRMWARE
2	SDK_VER_DRIVER
3	SDK_VER_LIBRARY
Callback function pointer types	
1	SDK_CB_GET_IRIS_IMAGE
2	SDK_CB_GET_LIVE_INFO
3	SDK_CB_GET_STATUS
Status event types	
1	SDK_EVENT_DEVICE_TILT
2	SDK_EVENT_SURPRISE_REMOVE
3	SDK_EVENT_TXN_TIMEOUT
4	SDK_EVENT_TXN_ERROR
Capture modes	
1	SDK_MODE_ENROLLMENT
2	SDK_MODE_RECOGNITION
Eye types	
1	SDK_EYE_RIGHT
2	SDK_EYE_LEFT
3	SDK_EYE_BOTH
4	SDK_EYE_EITHER
Capture types	
0	SDK_CAPTURE_TYPE_AUTO
1	SDK_CAPTURE_TYPE_MANUAL
Capture priority	
0	SDK_PRIORITY_SPEED
1	SDK_PRIORITY_QUALITY
2	SDK_PRIORITY_MULTIPLE
Device tilt status	
0	SDK_TILT_OFF
1	SDK_TILT_ON

Transaction error types	
1	SDK_ERR_TXN_IMAGE
2	SDK_ERR_TXN_ILLUMINATION
3	SDK_ERR_TXN_SHUTTER
4	SDK_ERR_TXN_SDK_ABORT
Misc	
0	DISABLED
1	ENABLED

7. Appendix

7.1 Device Control Definitions

7.1.1 iCAM T10 Device Error Codes

Code	Name	Description
0	SDK_ERR_SUCCESS	Success
101	SDK_ERR_NOT_INIT	SDK is not initialized
102	SDK_ERR_OPEN	Open error
103	SDK_ERR_PARAMETER	Invalid parameter
104	SDK_ERR_INVALID_STATE	SDK not in proper state
105	SDK_ERR_CB_EVENT	Error in callback mechanism
106	SDK_ERR_API_FAIL	Error in calling internal API
-1	SDK_ERR_UNKNOWN	Unknown error

8. Reference

Please read Readme.pdf for the reference.

9. Technical Support

Additional Information and Technical assistance is available on the Iris ID's support web site at www.irisid.com, click on Support & Service then Technical Support.