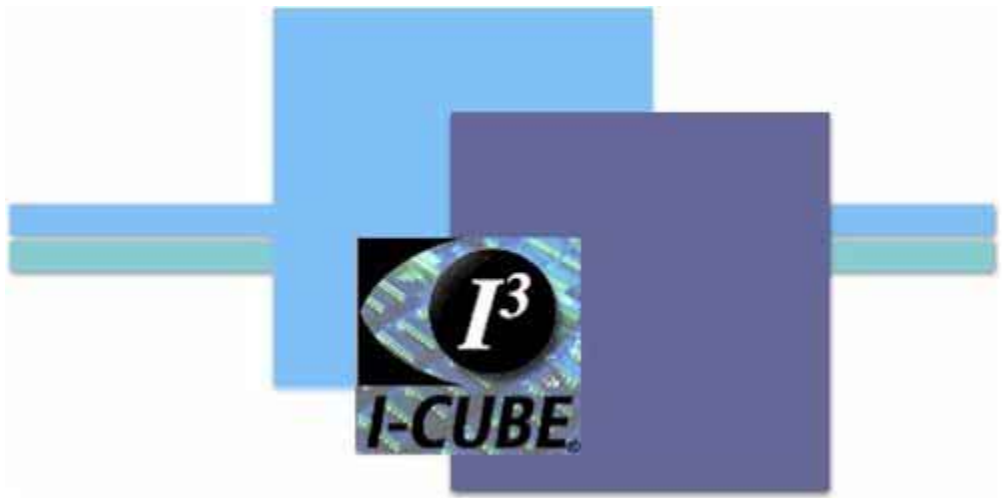


# FRS Discovery System OCX Control Reference Manual Beta Release – Mar 12/04





Copyright © 2004, Acsys Biometrics Corp. All Rights Reserved.

All information in this User's Manual is proprietary, except where indicated, and cannot be reproduced or redistributed without express written permission from Acsys Biometrics Corp.

Acsys Biometrics Corp does not guarantee the accuracy of the information contained in this document. Content is subject to change without notice.

Microsoft®, Windows®, and Windows® 2000 Professional®, and Windows® XP Professional are registered trademarks of Microsoft® Corporation.

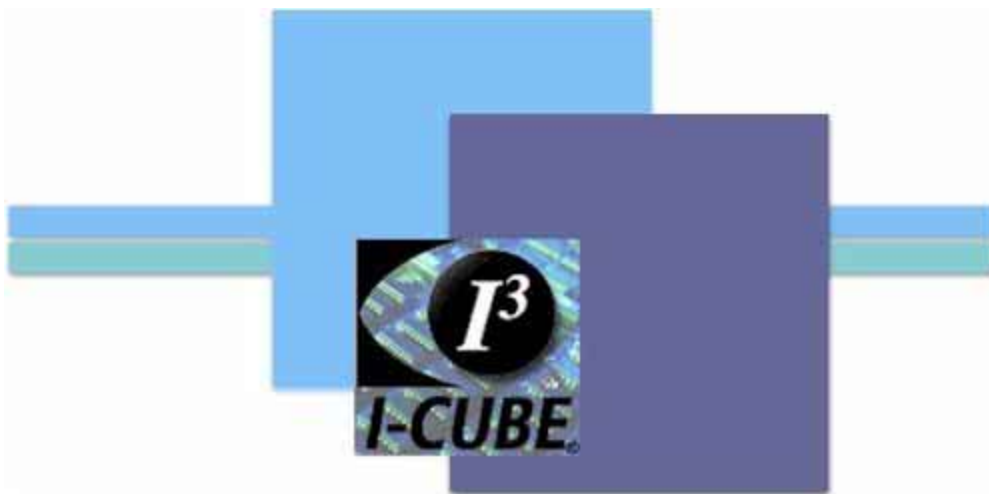
Intel®, and Pentium® are registered trademarks of Intel® Corporation.

InstallShield® is a registered trademark of InstallShield® Software Corporation.

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
<b>2</b>	<b>OCX METHODS.....</b>	<b>3</b>
2.1.1	ConnectClient.....	4
2.1.2	GetImageSegment .....	4
2.1.3	SetVideo .....	6
2.1.4	ShowDatabaseEdit.....	7
2.1.5	ShowEventViewer .....	8
2.1.6	ShowOptionsDialog .....	9
2.1.7	StartTrack.....	10
2.1.8	StopTrack .....	10
2.1.9	StartClassify.....	11
2.1.10	StopClassify .....	11
2.1.11	StartVerify .....	12
2.1.12	StopVerify.....	12
2.1.13	StartEnroll.....	13
2.1.14	StopEnroll.....	14
<b>3</b>	<b>OCX PROPERTIES.....</b>	<b>15</b>
3.1.1	ClassifyState .....	16
3.1.2	ClosestMatchConfidence .....	16
3.1.3	ClosestMatchID .....	17
3.1.4	ClosestMatchName .....	17
3.1.5	EnrollState .....	18
3.1.6	EnrollCount .....	18
3.1.7	HeadSize .....	19
3.1.8	HeadXaxis.....	19
3.1.9	HeadYangle .....	20
3.1.10	HeadYaxis.....	20
3.1.11	HeadZangle .....	21
3.1.12	ImageFrame .....	21
3.1.13	RecognitionConfidence .....	22
3.1.14	RecognitionID .....	22
3.1.15	RecognitionName .....	23
3.1.16	ShowClosestMatch.....	23
3.1.17	TrackConfidence.....	24
3.1.18	TrackCount .....	24
3.1.19	TrackEnum .....	25
3.1.20	TrackState .....	25

3.1.21	VerifyState .....	26
3.1.22	DisplayMode .....	26
<b>4</b>	<b>OCX EVENTS .....</b>	<b>27</b>
4.1.1	ClassifySuccess .....	28
4.1.2	VerifySuccess .....	28
4.1.3	EnrollSuccess .....	28
4.1.4	TrackFrame.....	29
4.1.5	TrackNew .....	29
4.1.6	TrackLost.....	29
4.1.7	ClassifyTimeout.....	30
4.1.8	VerifyTimeout .....	30
4.1.9	EnrollTimeout.....	30
4.1.10	ClientConnected .....	30



# 1 Introduction

The Discovery System OCX interface provides a general purpose application development platform for the Acsys FRS technology. This OCX interface allows the developer to integrate the FRS technology into a wide range of deployments including access control, monitoring and surveillance, time and attendance, and personnel security related fields. This Discovery system is based upon a Client-Server architecture, whereby the Server application generates and stores all biometric templates and these are subsequently distributed to one or more Client applications. The Client application (as well as Server) performs facial tracking, enrollment, verification and classification operations. Client applications are fully functional when disconnected from the Server.

The Discovery application illustrates the functionality of Acsys facial recognition within technology in a variety of practical applications, these being:

- 1) Facial enrollment from either live video stream input (i.e. camera, VCR, broadcast, etc.) or from static images (i.e. JPEG files).
- 2) Facial recognition in classification mode (one-to-many identification).
- 3) Facial recognition in verification mode (one-to-one identification).
- 4) Facial verification combined with Wiegand signal (i.e. proximity card, keypad, etc.).
- 5) Operation of R410/R810 relay controller following classification or verification operations (i.e. for opening door, controlling signs, etc.)
- 6) Interface for remote alarm on classify/verify operations (i.e. mobile wireless PDA, PECOS)
- 7) Speech synthesis on recognition (classification) of an individual.

A description of the OCX function interface for the Discovery System is provided in the following.

## 2 OCX METHODS

## 2.1 ConnectClient

This function is used to connect a Discovery Client application to a specific Discovery Server containing the FRS database and template generation modules. Once a client has been connected to a specific server, subsequent execution of the application will attempt reconnect to the same Discovery Server machine.

### Function Prototype

ConnectClient (ServerName As String)

ServerName The name of the computer on which the Discovery Server resides.

### Remarks

The function should be called only from Discovery Client machines.

## 2.2 GetImageSegment

Returns the facial image captured from the most recent video frame for the specified tracker thread. The segmented image that is extracted from the video capture frame is returned by this function in JPEG format. Also returned by this method are the coordinates of the face relative to the image segment that is returned.

### Function Prototype

GetImageSegment(trackerIndex As Long, ByRef ImageBuf As Variant,  
ByRef Coordinates As Variant)

trackerIndex Index to tracker thread. The valid index range is 0 to 15.

ImageBuf Returns a safearray of type BYTE containing the JPEG image of the tracked face.

Coordinates Returns a set of coordinate values that define the head position and tilt relative to the image segment returned by this function. In the event that this variable is assigned a NULL pointer, head coordinate values are not returned.

The following six floating point values are used to define head position and tilt:

HeadXaxis  
HeadYaxis  
HeadSize  
HeadXangle (not required)  
HeadYangle (not required)  
HeadZangle

## Remarks

This function returns the segmented facial image rather than the full capture frame that is returned by using **ImageFrame**. Tracking of facial objects is initiated using **StartTrack**; and images of the tracked faces may be retrieved on a frame-by-frame basis using **GetImageSegment** following a **TrackFrame** event.

Within the *Coordinates* parameter, the head coordinate and tilt values are expressed as follows:

HeadXaxis	center of tracked region as a fraction of the image segment width
HeadYaxis	center of tracked region as a fraction of the image segment height
HeadSize	height of region as a function of the image segment height
HeadXangle	degrees of head tilt along vertical normal (forward and back) * not used
HeadYangle	degrees of head tilt along Medula azimuth (sideways motion)
HeadZangle	degrees of head tilt along vertical tangent (shoulder to shoulder)

## See Also

FilterVariance, ImageFrame, StartTrack

## 2.3 SetVideo

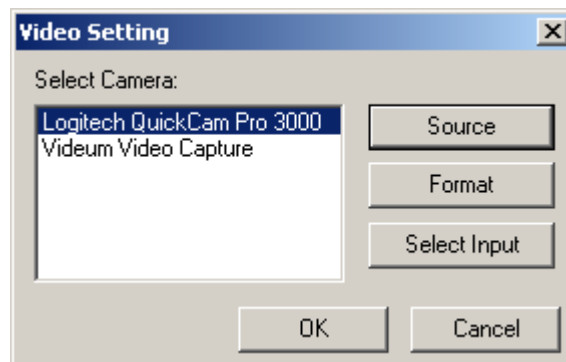
Displays the dialog box used for setting video properties.

### Function Prototype

SetVideo()

### Remarks

The video dialog box that is displayed is shown below, and the buttons activate the corresponding form for the selected video driver as described below.



### Source

This pushbutton activates a properties dialog for the selected video driver, which generally provides the following adjustments. The form layout and options vary considerably between video capture providers.

- Brightness
- Contrast
- Gamma
- Saturation
- Auto Exposure
- Frame Rate

## Format

This pushbutton activates a second properties dialog for the selected video driver. The various video drivers generally provides the following adjustments:

Compression  
Pixel Resolution

In the event that video has not been initialized using the **Initialize** function, **SetVideo** will initialize video

## 2.4 ShowDatabaseEdit

Displays the Discovery database edit screen.

### Function Prototype

ShowDatabaseEdit( )

### Remarks

The Discovery database edit screen allows the user to view and delete images and users within the FRS database. Menu items provided in this screen also allow the user to import and export individuals between separate installations of the FRS Discovery system. Further information regarding the Discovery database edit screen is provided in Section 3.3 of the Discovery manual. The database edit screen is available on the Discovery Server only.

## 2.5 ShowEventViewer

Displays the Discovery event viewer screen.

### Function Prototype

ShowEventViewer( )

### Remarks

The Discovery event viewer allows the user to browse event information. This information includes the following activities.

- Enroll Failure (Face)
- Enroll Success
- Reenroll
- Verify Failure (Face)
- Verify Success
- Face Classified
- Face Unknown

The information displayed for each event is listed as follows:

- Log ID number
- Time and Date
- User Name
- Client (computer) name
- Activity Type

Facilities provided on the event viewer screen provide the following capabilities:

- Multiple document display of event lists
- Import/Export of events
- Printing of event lists
- Copying of the JPEG image recorded for each event
- Filtering of events

Further information regarding the Discovery event viewer screen is provided in Section 3.5 of the Discovery manual. The event viewer screen is available on the Discovery Server only.

## 2.6 ShowOptionsDialog

Displays the Discovery Options dialog. The Options dialog allows the user to modify operational settings for the Discovery System.

### Function Prototype

ShowOptionsDialog( )

### Remarks

The options dialog varies slightly between Client and Server installations. The following groups of settings provided within separate tabs of the dialog are provided for the Discovery System, as follows

Client Machine	Server Machine
General	General
Enrollment	Enrollment
Tracking	Tracking
Classify	Classify
Verify	Verify
Image Filter	Image Filter
Extensions	Extensions
	Training

Further information regarding the Discovery options dialog is provided in Section 3.2 of the Discovery manual.

## 2.7 StartTrack

Initiates the facial tracking operation. This operation is identical to that operation performed by manually clicking the "Track" button provided on the Discovery GUI.

### Function Prototype

```
StartTrack( )
```

### Remarks

FRS Discovery System allows tracking of up to 16 individuals simultaneously. The maximum number of tracking threads is adjustable through the Options dialog.

## 2.8 StopTrack

Terminates the tracking operation. This operation is identical to clicking the "Track" button provided on the Discovery GUI while tracking is active.

### Function Prototype

```
StopTrack( )
```

### Remarks

On initiation of a verify, classify or enroll event, the tracking operation is started automatically. The tracking operation, however, is not terminated when the verify or enroll is complete or when the classify operation is terminated. This function may be used to terminate the tracking operation following verify, classify or enroll.

## 2.9 StartClassify

Initiates the classification operation. This operation is identical to clicking the "Classify" button provided on the Discovery GUI.

### Function Prototype

```
StartClassify( )
```

### Remarks

The classify operation performs a one-to-many identification. In the event that the classify thresholds are exceeded the **ClassifyEvent** is triggered. The individuals identity may subsequently be returned using **RecognitionID** and **RecognitionName** and the recognition confidence is returned using the **RecognitionConfidence** property. The set of closest matches to the tracked individual (compared against those individuals currently enrolled within the FRS database) may be returned using **ClosestMatchesID** and **ClosestMatchesName**.

## 2.10 StopClassify

Terminates the classify operation. This operation is identical to clicking the "Classify" button provided on the Discovery GUI while the classify operation is active.

### Function Prototype

```
StopClassify( )
```

### Remarks

Following termination of the classify operation the tracking operation will remain active. The tracking operation may be terminated using the **StopTrack** method.

## 2.11 StartVerify

Initiates the verify operation. This operation is identical to clicking the "Verify" button provided on the Discovery GUI. The option is provided to enter user identity through the function parameter list, or activate the verify dialog for manual entry of user identity.

### Function Prototype

StartVerify(DialogFlag As Boolean, UserID As String)

DialogFlag    Determines if user ID is entered using the function parameter UserID or entered manually through the verify dialog, as follows:

TRUE            User ID entered using the verify dialog

FALSE           User ID entered through the function parameter

UserID         User ID number for the individual that is being verified

### Remarks

The verify operation performs one-to-one identification. In the event that the verify threshold is exceeded the **VerifyEvent** is triggered. The individual's identity may subsequently be returned using **RecognitionID** and **RecognitionName** and the recognition confidence is returned using **RecognitionConfidence**.

## 2.12 StopVerify

Terminates the verify operation. The verify operation is terminated automatically if either the verify threshold is exceeded or a verify timeout occurs.

### Function Prototype

StopVerify( )

### Remarks

Following termination of the verify operation the tracking operation will remain active. The tracking operation may be terminated using the **StopTrack** method.

## 2.13 StartEnroll

Initiates the enrollment operation. This operation is identical to clicking the "Enroll" button provided on the Discovery GUI.

### Function Prototype

StartEnroll(DialogFlag As Boolean, FilterFlag as Boolean, UserID As String,  
                  FirstName as String, LastName As String,  
                  maxCount As Long)

DialogFlag    Determines if user ID is entered using the function parameter UserID  
                  or entered manually through the verify dialog, as follows:

                  TRUE            User ID and name entered using the enroll dialog

                  FALSE          User ID and name entered through the function  
                                  parameter list

FilterFlag    Determines if the filter dialog (used for previewing and deleting  
                  enrolled images) is activated following the enrollment operation, as  
                  follows:

                  TRUE            Filter dialog is enabled following enrollment operation

                  FALSE          Filter dialog is not enabled

UserID        User ID number for the individual that is being enrolled

FirstName    First name of the user that is being enrolled

LastName     Last name of user that is being enrolled

maxCount     Sets the maximum number of images that are enrolled

### Remarks

In the event that the filter dialog is disabled, enrollment images may be manually reviewed and filtered using the edit database screen provided on the Discovery Server.

## 2.14 StopEnroll

Terminates the enrollment operation. The enrollment operation is terminated automatically if an enroll timeout occurs or when the specified quantity enrollment images have been captured.

### Function Prototype

StopEnroll(AbortFlag As Long)

AbortFlag This flag determines whether those images captured prior to termination are discarded or stored within the FRS database, as follows:

- |   |   |
|---|---|
| 1 | Discard images and all related user information from FRS database |
| 0 | Retain captured images and user information                       |

### Remarks

Following termination of the verify operation the tracking operation will remain active. The tracking operation may be terminated using the **StopTrack** method.

## 3 OCX PROPERTIES

### 3.1 ClassifyState

Returns the current state of the classify operation. This property is read only.

#### Function Prototype

ClassifyState( ) As Boolean

#### Remarks

This function returns a Boolean flag that indicates the current state of the classify operation, as follows:

TRUE	Classify is currently active
FALSE	Classify is not currently active

### 3.2 ClosestMatchConfidence

Returns the recognition confidence for the top N closest matches sorted in decreasing order of recognition confidence (classify operation only). These values are returned for the specified tracker thread. This property is read only.

#### Function Prototype

ClosestMatchConfidence ( trackerIndex as Long, matchCount As Long)  
As Variant

trackerIndex    Index to tracker thread. The valid index range is 0 to 15.

matchCount    Specifies the number of closest matches that are returned

#### Remarks

Returned is a safearray of type Single containing the recognition confidence values for the top N matches. In the event that the specified tracker thread is not currently tracking a face, a NULL array is returned.

### 3.3 ClosestMatchID

Returns the user ID numbers for the top N closest matches sorted in decreasing order of recognition confidence (classify operation only). These values are returned for the specified tracker thread. This property is read only.

#### Function Prototype

ClosestMatchID ( trackerIndex as Long, matchCount As Long ) As Variant

trackerIndex     Index to tracker thread. The valid index range is 0 to 15.

matchCount     Specifies the number of closest matches that are returned

#### Remarks

Returned is a safearray of type String containing the user ID numbers for the top N matches. In the event that the specified tracker thread is not currently tracking a face, a NULL array is returned.

### 3.4 ClosestMatchName

Returns the user first and last name for the top N closest matches sorted in decreasing order of recognition confidence (classify operation only). These values are returned for the specified tracker thread. This property is read only.

#### Function Prototype

ClosestMatchName( trackerIndex as Long, matchCount As Long ) As Variant

trackerIndex     Index to tracker thread. The valid index range is 0 to 15.

matchCount     Specifies the number of closest matches that are returned

#### Remarks

Returned is a safearray of type String containing the user name for the top N matches. The string returned contains the first name followed by last name using a comma separator. In the event that the specified tracker thread is not currently tracking a face, a NULL array is returned.

### 3.5 EnrollState

Returns the current state of the enrollment operation. This property is read only.

#### Function Prototype

EnrollState( ) As Boolean

#### Remarks

This function returns a Boolean flag that indicates the current state of the enroll operation, as follows:

TRUE     Enrollment is currently active

FALSE    Enrollment is not currently active

### 3.6 EnrollCount

Returns the current number of facial images that have been captured during an enrollment operation. This property is read only.

#### Function Prototype

EnrollCount(trackerIndex As Long) As Long

trackerIndex     Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

Returns a long containing the number of images that have been enrolled from the current enrollment operation.

### 3.7 HeadSize

This property returns the size of the tracker box for the specified tracker thread. This property is read only.

#### Function Prototype

HeadSize(trackerIndex As Long) As Single

trackerIndex    Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

The size is expressed as the height of the tracker box divided by total image frame height. If a face is not being tracked by the specified thread a value of 0.0 is returned.

### 3.8 HeadXaxis

This property returns the horizontal coordinate value of the specified tracker thread. This property is read only.

#### Function Prototype

HeadXaxis(trackerIndex As Long) As Single

trackerIndex    Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

The horizontal coordinate is expressed as the distance from the left side of the frame divided by total frame width. The coordinate value returned corresponds to the center of the tracking box. If the specified tracking thread has not currently located a face, a value of 0.0 is returned.

### 3.9 HeadYangle

This property returns the degree of head tilt in the azimuthal direction (sideways motion) for the specified tracker thread. This property is read only.

#### Function Prototype

HeadYangle(trackerIndex As Long) As Single

trackerIndex    Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

This value is expressed in degrees with 0 degrees corresponding to the full frontal projection. If the specified tracking thread has not currently located a face, a value of 0.0 is returned.

### 3.10 HeadYaxis

This property returns the vertical coordinate value of the specified tracker thread. This property is read only.

#### Function Prototype

HeadYaxis(trackerIndex As Long) As Single

trackerIndex    Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

The vertical coordinate is expressed as the distance from the top of the frame divided by total frame height. The coordinate value returned corresponds to the center of the tracking box. If the specified tracking thread has not currently located a face, a value of 0.0 is returned.

### 3.11 HeadZangle

This property returns the degree of head tilt in the shoulder to shoulder direction for the specified tracker thread. This property is read only.

#### Function Prototype

HeadZangle(trackerIndex As Long) As Single

trackerIndex     Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

This value is expressed in degrees with 0 degrees corresponding to the head oriented in the vertical direction. If the specified tracking thread has not currently located a face, a value of 0.0 is returned.

### 3.12 ImageFrame

Returns a JPEG image of the full frame for the current capture device. This property is read only.

#### Function Prototype

ImageFrame() As Variant

#### Remarks

This function executes on the client module and the image returned applies to the specified capture device. Returned is a safearray of type BYTE containing the JPEG image of the full capture frame.

### 3.13 RecognitionConfidence

Returns the recognition confidence value for specified tracker thread during a classify or verify operation. This property is read only.

#### Function Prototype

RecognitionConfidence (trackerIndex as Long) As Single

trackerIndex    Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

The current recognition confidence value is returned for either a verify or classify operation. The default threshold applied for verification or classification operations is 0.4. The valid range for these thresholds are from 0.0 (low confidence) to 1.0 (high confidence). In the event that a classify operation is being performed, this property returns the largest confidence value among all users that have been enrolled within the system. In the event that the specified tracker thread is not currently actively tracking a face, a value of -1.0 is returned.

### 3.14 RecognitionID

Returns user ID number that corresponds to the largest recognition confidence value during a classify or verify operation. This ID number is returned for the specified tracker thread. This property is read only.

#### Function Prototype

RecognitionID( trackerIndex as Long) As String

trackerIndex    Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

In the event that the specified tracker thread is not currently tracking a face, Null string is returned.

### 3.15 RecognitionName

Returns the user first and last name that corresponds to the largest recognition confidence value during a classify or verify operation. The user name is returned for the specified tracker thread. This property is read only.

#### Function Prototype

RecognitionName( trackerIndex as Long ) As String

trackerIndex     Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

The user name is returned by first name followed by last name using a comma separator. In the event that the specified tracker thread is not currently tracking a face, a NULL string is returned.

### 3.16 ShowClosestMatch

Specifies the display state of the closest matches overlay. This overlay is shown on the right boarder of the video display and can be enabled only during a classify operation. This operation is identical to clicking the "Closest Matches" button provided on the Discovery GUI.

#### Function Prototype

ShowClosestMatch() As Boolean

#### Remarks

The display state for the closest matches overlay is provided as follows:

TRUE     Closest matches overlay is displayed

FALSE    Closest matches overlay is not displayed

### 3.17 TrackConfidence

Returns the tracking confidence for the specified tracker thread. This property is read only.

#### Function Prototype

TrackConfidence(trackIndex As Long) As Single

trackIndex                      Index to tracker thread. The valid index range is 0 to 15.

#### Remarks

The tracking confidence measures the level of confidence that the object is in fact a face. The default confidence threshold applied for facial tracking is 0.4.

### 3.18 TrackCount

Returns the number of tracker threads that are currently active (i.e. tracking faces). This property is read only.

#### Function Prototype

TrackCount() As Long

#### Remarks

The number of tracking threads is dependent upon the number of individuals that are currently being tracked by the system. For instance if 3 individuals are currently tracked, the number of active tracking threads will be 3 + 1 (used for locating new individuals). The user may adjust the maximum number of tracking threads deployed by adjusting the tracking properties within the Options dialog.

### 3.19 TrackEnum

Returns the thread ID numbers for the tracker threads that are operational in currently tracking faces. This property is read only.

#### Function Prototype

TrackCount() As Variant

#### Remarks

The number of tracking threads is dependent upon the number of individuals that are currently being tracked by the system. For instance if 3 individuals are currently tracked, the number of active tracking threads will be 3 (does not include the free thread that is used for locating a new individual).

The number of active tracking threads is determined by **TrackCount**. The **TrackEnum** property returns a safearray of type Long containing the thread ID numbers.

### 3.20 TrackState

Returns the current state of the tracking operation. This property is read only.

#### Function Prototype

TrackState( ) As Boolean

#### Remarks

This function returns a Boolean flag that indicates the current state of the tracking operation, as follows:

TRUE     Tracking is currently active

FALSE    Tracking is not currently active

### 3.21 VerifyState

Returns the current state of the classify operation. This property is read only.

#### Function Prototype

VerifyState( ) As Boolean

#### Remarks

This function returns a Boolean flag that indicates the current state of the verify operation, as follows:

TRUE     Verify is currently active

FALSE    Verify is not currently active

### 3.22 DisplayMode

Two video modes are currently supported. These are display of Discovery GUI application in full screen mode, or display of video and tracking icon only with no GUI interface.

#### Function Prototype

DisplayMode() As Long

#### Remarks

This function is used to set or return a Long, which establishes the display mode for the OCX control. The three options provided are listed below:

0            Display of captured video with tracking overlay and Discovery GUI shown in default full screen.

1            Display of captured video with tracking overlay is enabled, however display of the associated Discovery GUI is disabled. The video display is sized to fill the control area and the display state is enabled or disabled using the OCX control show/hide properties.

## 4 OCX EVENTS

## 4.1 ClassifySuccess

Event triggered when a classification operation is active and the facial image returned from any tracker thread exceeds the classify thresholds. This event returns the tracker thread index and user ID number.

### Function Prototype

ClassifySuccess (trackerIndex As Long, UserID As String)

trackerIndex     Index for the tracker thread that exceeded the classify threshold

UserID            The ID number for the associated user

## 4.2 VerifySuccess

Event triggered when a verification operation is active and the facial image returned from any tracker thread exceeds the verify threshold. This event returns the tracker thread index and user ID number.

### Function Prototype

VerifySuccess (trackerIndex As Long, UserID As String)

trackerIndex     Index for the tracker thread that exceeded the classify threshold

UserID            The ID number for the associated user

## 4.3 EnrollSuccess

Event triggered when an enrollment operation has been completed. This event is generated only in the event that images have been captured and are saved for transfer to the FRS database, regardless of whether the enrollment is terminated as a result of an enrollment timeout condition or programmatically terminated through the use of the **StopEnroll** method.

### Function Prototype

EnrollSuccess ( )

## 4.4 TrackFrame

Returns an event if tracking is active through a call to **StartTrack** and a face has been located within the current frame. This event is produced for every frame in which a face has been located.

### Function Prototype

TrackFrame(trackerIndex As Long)

trackerIndex    Index to tracker thread which located a face.

## 4.5 TrackNew

Returns an event when a tracker thread has located a face within a video stream following a call to **StartTrack**, **StartClassify**, **StartVerify** or **StartEnroll**. This event is produced for the first frame in which a tracking thread has located a face.

### Function Prototype

TrackNew(trackerIndex As Long)

trackerIndex    Index to the tracker thread which initiated tracking of a face.

## 4.6 TrackLost

Returns an event if a tracker thread has lost tracking of the face within a video stream following a call to **StartTrack**, **StartTrack**, **StartClassify**, **StartVerify** or **StartEnroll**. This event is produced for the first frame in which a tracking thread has lost tracking of a face.

### Function Prototype

TrackLost(trackerIndex As Long)

trackerIndex    Index to the tracker thread which lost tracking of a face.

## 4.7 ClassifyTimeout

Returns an event when a classify operation has been terminated due to a classify timeout condition.

### Function Prototype

ClassifyTimeout()

## 4.8 VerifyTimeout

Returns an event when a verify operation has been terminated due to a verify timeout condition.

### Function Prototype

VerifyTimeout( )

## 4.9 EnrollTimeout

Returns an event when an enroll operation has been terminated due to an enroll timeout condition.

### Function Prototype

EnrollTimeout( )

## 4.10 ClientConnected

Returns a event when client has been connected to a new FRS server (ClientConnect function succeeded)

### Function Prototype

ClientConnected (ServerName As String )

ServerName – the name of server.