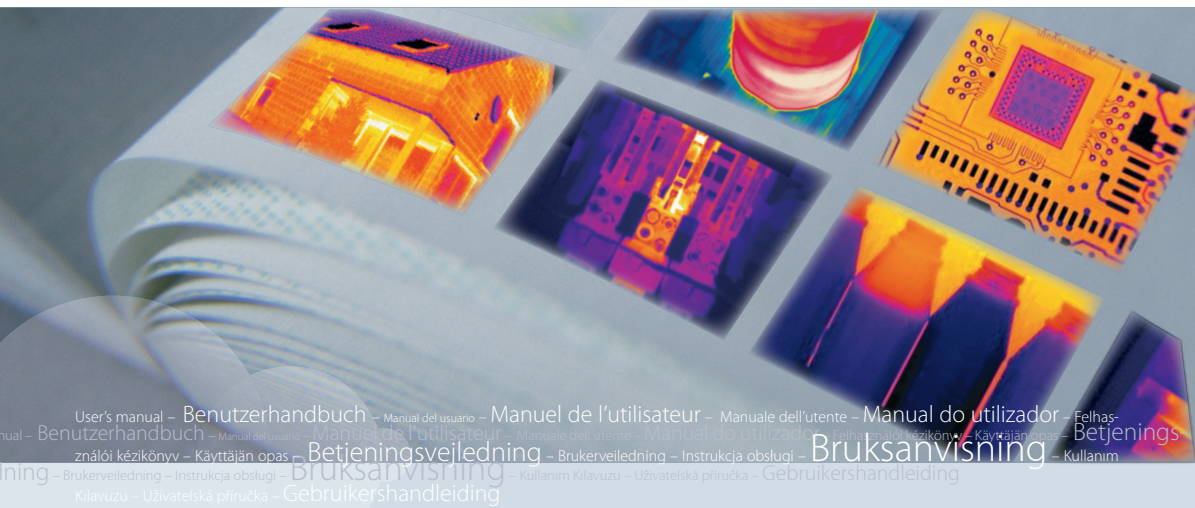




User's Manual



User's manual – Benutzerhandbuch – Manual del usuario – Manuel de l'utilisateur – Manuale dell'utente – Manual do utilizador – Felhasználói kézikönyv – Käyttöäjan opas – Betjeningsveiledning – Brukerveiledning – Instrukcja obsługi – Bruksanvisning – Kullanim Kilavuzu – Uživatelská příručka – Gebruikershandleiding

OLE Automation

For ThermaCAM™ Researcher 2.7, 2.8 & 2.9

Publ. No.	T559011
Revision	a318
Language	English (EN)
Issue date	February 20, 2009

Customer help

1

About OLE Automation programming

2

Properties

3

Methods

4

Index

5

OLE Automation

User's Manual



Legal disclaimer

All products manufactured by FLIR Systems are warranted against defective materials and workmanship for a period of one (1) year from the delivery date of the original purchase, provided such products have been under normal storage, use and service, and in accordance with FLIR Systems instruction.

All products not manufactured by FLIR Systems included in systems delivered by FLIR Systems to the original purchaser carry the warranty, if any, of the particular supplier only and FLIR Systems has no responsibility whatsoever for such products.

The warranty extends only to the original purchaser and is not transferable. It is not applicable to any product which has been subjected to misuse, neglect, accident or abnormal conditions of operation. Expendable parts are excluded from the warranty.

In the case of a defect in a product covered by this warranty the product must not be further used in order to prevent additional damage. The purchaser shall promptly report any defect to FLIR Systems or this warranty will not apply.

FLIR Systems will, at its option, repair or replace any such defective product free of charge if, upon inspection, it proves to be defective in material or workmanship and provided that it is returned to FLIR Systems within the said one-year period.

FLIR Systems has no other obligation or liability for defects than those set forth above.

No other warranty is expressed or implied. FLIR Systems specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

FLIR Systems shall not be liable for any direct, indirect, special, incidental or consequential loss or damage, whether based on contract, tort or any other legal theory.

Copyright

© FLIR Systems, 2009. All rights reserved worldwide. No parts of the software including source code may be reproduced, transmitted, transcribed or translated into any language or computer language in any form or by any means, electronic, magnetic, optical, manual or otherwise, without the prior written permission of FLIR Systems.

This manual must not, in whole or part, be copied, photocopied, reproduced, translated or transmitted to any electronic medium or machine readable form without prior consent, in writing, from FLIR Systems.

Names and marks appearing on the products herein are either registered trademarks or trademarks of FLIR Systems and/or its subsidiaries. All other trademarks, trade names or company names referenced herein are used for identification only and are the property of their respective owners.

Quality assurance

The Quality Management System under which these products are developed and manufactured has been certified in accordance with the ISO 9001 standard.

FLIR Systems is committed to a policy of continuous development; therefore we reserve the right to make changes and improvements on any of the products described in this manual without prior notice.

Table of contents

1	Customer help	1
2	About OLE Automation programming	3
2.1	Obtaining an OLE Automation object reference	3
3	Properties	5
3.1	AmbientTemperature Property	5
3.2	AnalysisDistance Property	5
3.3	AnalysisEmissivity Property	5
3.4	AnalysisLabel Property	6
3.5	AnalysisTemperature Property	6
3.6	AtmosphericTemperature Property	7
3.7	AutoMaxTemperature Property	8
3.8	AutoMinTemperature Property	8
3.9	BBRange Property	9
3.10	CalcWhileMovingAna Property	9
3.11	Camera Property	9
3.12	CameraFilter Property	10
3.13	CameraLens Property	10
3.14	CameraLevel Property	10
3.15	CameraRangeMax Property	11
3.16	CameraRangeMin Property	11
3.17	CameraSerialNo Property	11
3.18	CameraSpan Property	12
3.19	CameraType Property	12
3.20	CanHaveBarCodes Property	12
3.21	Cassette Property	13
3.22	ComPort Property	13
3.23	Connected Property	13
3.24	Cooler Property	14
3.25	CurrentIROutput Property	14
3.26	DefaultHistogramClasses Property	15
3.27	Device Property	15
3.28	DistanceUnit Property	15
3.29	Emissivity Property	16
3.30	EstimatedTransmission Property	16
3.31	ExtOpticsTemperature Property	16
3.32	ExtOpticsTransmission Property	17
3.33	Filter Property	17
3.34	Filtering Property	17
3.35	Freeze Property	18
3.36	ImageDate Property	18
3.37	ImageFilename Property	18
3.38	ImageMaxTemperature Property	19
3.39	ImageMinTemperature Property	19
3.40	ImagePath Property	20
3.41	ImageTime Property	20
3.42	ImageTimeMSec Property	20
3.43	ImageTimeStr Property	21
3.44	ImageTitle Property	21

3.45	IsFirstImage Property	21
3.46	IsLastImage Property	22
3.47	IsUpdated Property	22
3.48	KeepAnalysis Property	22
3.49	Motor Property	23
3.50	ObjectDistance Property	23
3.51	OLELiveLinkDelay Property	23
3.52	PreferredIROutput Property	24
3.53	Ratio Property	24
3.54	RecAutoNameBase Property	24
3.55	ReclmgPres Property	25
3.56	RecOneFileRec Property	25
3.57	RecStartCond Property	25
3.58	RecStartDelay Property	26
3.59	RecStartTime Property	26
3.60	RecStopCond Property	27
3.61	RecStopDelay Property	27
3.62	RecStopDuration Property	27
3.63	RecStopImages Property	28
3.64	RecStoreCond Property	28
3.65	RecStoreImages Property	29
3.66	RecStoreInterval Property	29
3.67	ReferenceTemperature Property	29
3.68	RefTemperatureExists Property	30
3.69	RelativeHumidity Property	30
3.70	Save900Analysis Property	30
3.71	ShowAnalysis Property	31
3.72	ShowAnalysisBackColor Property	31
3.73	ShowAnalysisTemp Property	32
3.74	ShowIlegTempIndicator Property	32
3.75	ShowLabels Property	33
3.76	ShowTemperatureScale Property	33
3.77	TargetMotion Property	33
3.78	TemperatureUnit Property	34
3.79	TotalImages Property	34
3.80	UselImageObjParams Property	35
3.81	UselImageScale Property	35
3.82	UselImageZoom Property	35
3.83	Version Property	36
3.84	View3D Property	36
3.85	Visible Property	36
3.86	Zoomfactor Property	37
4	Methods	39
4.1	AddPlaybTextCmt Method	39
4.2	AddRecTextCmt Method	39
4.3	AnaDefineFormula Method	40
4.4	AnaDeleteFormula Method	40
4.5	AnalysisDefineBentLine Method	41
4.6	AnalysisDefineBoxArea Method	42
4.7	AnalysisDefineCircleArea Method	42
4.8	AnalysisDefineLine Method	43
4.9	AnalysisDefinePolygonArea Method	44

4.10	AnalysisDefineSpotmeter Method	44
4.11	AnalysisDeleteFunc Method	45
4.12	AnalysisFuncNames Method	45
4.13	AnalysisHistoArray Method	46
4.14	AnalysisHistoClasses Method	47
4.15	AnalysisHistoThreshold Method	47
4.16	AnalysisResults Method	48
4.17	AnalysisTemperatureArray Method	49
4.18	AnalysisTemperatureArrayLen Method	49
4.19	AnalysisTemperatureByName Method	50
4.20	AutoAdjust Method	51
4.21	AutoRewind Method	51
4.22	Connect Method	51
4.23	CopySessionAndImage Method	52
4.24	Disconnect Method	52
4.25	ExternalCorrection Method	52
4.26	FastBackward Method	52
4.27	FastForward Method	53
4.28	FocusFar Method	53
4.29	FocusNear Method	53
4.30	FocusStop Method	53
4.31	GetFormulaByIndex Method	54
4.32	GetFormulaByName Method	54
4.33	GetFrameRate Method	55
4.34	GetNamedValue Method	55
4.35	GetNoFormulas Method	55
4.36	GetNumPlaybTextCmts Method	56
4.37	GetNumRecTextCmts Method	56
4.38	GetPaletteInfo Method	56
4.39	GetPlaybTextCmt Method	57
4.40	GetPlaybTextCmtByName Method	57
4.41	GetPlayRate Method	58
4.42	GetRecTextCmt Method	58
4.43	GetRecTextCmtByName Method	58
4.44	GetScaling Method	59
4.45	GotoFirstImage Method	59
4.46	GotoLastImage Method	60
4.47	ImageDimensions Method	60
4.48	InternalCorrection Method	60
4.49	KeyTrigF5 Method	61
4.50	LoadImage Method	61
4.51	NewSession Method	61
4.52	OpenSession Method	61
4.53	PlayBackward Method	62
4.54	PlayForward Method	62
4.55	PlayStop Method	62
4.56	RecordOneImage Method	63
4.57	Refresh Method	63
4.58	RemoveAllAnalysis Method	63
4.59	RemovePlaybTextCmt Method	64
4.60	RemovePlaybTextCmtByName Method	64
4.61	RemoveRecTextCmt Method	64
4.62	RemoveRecTextCmtByName Method	65

4.63	RestoreAnalysis Method	65
4.64	RestoreObjParameters Method	65
4.65	SaveAreaAsCsv Method	65
4.66	SaveImage Method	66
4.67	SaveSession Method	67
4.68	SetCameraSource Method	67
4.69	SetFrameRate Method	67
4.70	SetInverted Method	68
4.71	SetPalette Method	68
4.72	SetPlayRate Method	68
4.73	SetScaling Method	69
4.74	StepBackward Method	69
4.75	StepForward Method	70
4.76	SubtrGetResultImage Method	70
4.77	SubtrSaveOutput Method	70
4.78	SubtrSetOptions Method	71
4.79	SubtrSetOutput Method	71
4.80	SubtrSetOutputDir Method	72
4.81	SubtrSetReference Method	72
4.82	SubtrSetScale Method	73
4.83	SubtrSetSource Method	73
	Index	75

General

For customer help, visit:

<http://flir.custhelp.com/app/home>

Submitting a question

To submit a question to the customer help team, you must be a registered user. It only takes a few minutes to register online. If you only want to search the knowledge-base for existing questions and answers, you do not need to be a registered user.

When you want to submit a question, make sure that you have the following information to hand:

- The camera model
 - The camera serial number
 - The communication protocol, or method, between the camera and your PC (for example, Ethernet, USB™, or FireWire™)
 - Operating system on your PC
 - Microsoft® Office version
 - Full name, publication number, and revision number of the manual
-

Downloads

On the customer help site you can also download the following:

- Firmware updates for your infrared camera
- Program updates for your PC software
- User documentation
- Application stories
- Technical publications

1 Figure This figure shows the welcome page of FLIR Systems customer help site:

10776203.a3



2 About OLE Automation programming

With OLE Automation, ThermaCAM™ Researcher functionality is made available to e.g. a VBA script in an Excel module. For this purpose, ThermaCAM™ Researcher exposes an *OLE Automation* object. Other applications can create such objects and give them two types of commands: Properties (whose values can be read and changed) and Methods (that can be called with or without arguments).

An application that exposes its OLE Automation objects is called an *OLE server application*. An application that can access and manipulate OLE Automation objects is called an *OLE controller application*.

Before a controller application can exchange data with the server application it must create a reference to it.

2.1 Obtaining an OLE Automation object reference

There are two ways to refer to the OLE Automation objects of an application. One way is to refer to *the application's object library*. An object library contains a description of all the objects an application provides, including definitions of all available methods and properties.

To establish a reference to an application's object library in e.g. Microsoft Excel take the following actions:

- Switch to a Visual Basic Module.
- On the Tools menu, click References.
- Click the object library you want to refer to.

If the object library you want doesn't appear in the Available References box, you can use the Browse button to search for the object library file containing the object you need. Object libraries usually have a .tlb or an .olb filename extension. The ThermaCAM™ Researcher object library is located in the same directory as the ThermaCAM™ Researcher executable file.

After you've established a reference to an object library, you can use its commands from the library in Microsoft Excel Visual Basic code.

Another way to refer to another application is either to use the **CreateObject** method to start the application and return a reference to an object or to use the **GetObject** method to return a reference to an object from an applications that is already running. See your VBA documentation for a complete description of these methods.

INTENTIONALLY LEFT BLANK

3 Properties

3.1 *AmbientTemperature Property*

Set or get the image scale ambient temperature value

Syntax

object.AmbientTemperature = *temperature*

temperature = *object.AmbientTemperature*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>temperature</i>	Float. Ambient temperature.	Kelvin

3.2 *AnalysisDistance Property*

Set or get the distance for the specified analysis tool.

Syntax

object.AnalysisDistance(AnalysisName) = *distance*

distance = *object.AnalysisDistance(AnalysisName)*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>AnalysisName</i>	The analysis name	For example SP01,AR03,...
<i>distance</i>	Double: The analysis distance	

3.3 *AnalysisEmissivity Property*

Set or get the emissivity for the specified analysis tool.

Syntax

object.AnalysisEmissivity(AnalysisName) = *emissivity*

emissivity = *object.AnalysisEmissivity(AnalysisName)*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

Item	Description	Value
<i>AnalysisName</i>	The analysis name	For example SP01,AR03,...
<i>emissivity</i>	Double: The analysis emissivity	

3.4 *AnalysisLabel Property*

Set or get the label for the specified analysis tool.

Syntax

object.**AnalysisLabel**(*AnalysisTool*, *AnalysisNumber*) = *label*

label = *object*.**AnalysisLabel**(*AnalysisTool*, *AnalysisNumber*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>AnalysisTool</i>	Type of analysis symbol	1 Spotmeter 2 Area 3 Line 4 Isotherm
<i>AnalysisNumber</i>	The number of the symbol	1 For SP01 and AR01 etc.
<i>label</i>	The label for the analysis symbol	

NOTE: These generic names will not be changed by this method. You should still refer to spotmeter 1 as SP01 even if you change its label to something else.

3.5 *AnalysisTemperature Property*

Get a temperature value for an analysis function.

Syntax

temperature = *object*.**AnalysisTemperature**(*AnalysisType*, *AnalysisNumber*, *AnalysisValue*, *TempState*)

Item	Description	Value
<i>object</i>	The name of the ThermoCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>AnalysisType</i>	Type of analysis symbol.	1 Spotmeter (SPxx) 2 Area (ARxx) 3 Line (Llxx) 4 Isotherm (ISOxx)
<i>AnalysisNumber</i>	The number of the symbol	1 For SP01 and AR01 etc.
<i>AnalysisValue</i>	Type of temperature requested.	1 Temperature. Spotmeter, Isotherm 2 Min temperature. Line, Area 3 Max temperature. Line, Area 4 Average temperature. Line, Area 5 Cursor temperature. Line 6 Temperature width. Isotherm 7 Difference temperature. Spotmeter, Isotherm 8 Min difference temperature. Line, Area 9 Max difference temperature. Line, Area 10 Average difference temperature. Line, Area 11 Cursor difference temperature. Line
<i>TempState</i>	Passed by reference. The state of the temperature will be returned here.	0 OK 1 Warning, the temperature is unreliable 2 Underflow, too low temperature 3 Overflow, too high temperature 4 Invalid, the temperature could not be calculated
<i>temperature</i>	The temperature for the specified analysis function. If <i>TempState</i> is Invalid, the temperature can't be returned	

NOTE: The temperature is returned in the current temperature unit as double value. You can only retrieve temperature values, not set the values. If the analysis function does not exist, the return value is undefined.

NOTE: You can't expect the temperature to be correct until the ThermaCAM™ Researcher has updated itself completely to the image.

3.6 AtmosphericTemperature Property

Sets or gets the image scale atmospheric temperature value

Syntax

object.**AtmosphericTemperature** = *temperature*

temperature = *object*.**AtmosphericTemperature**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>temperature</i>	Float. Atmospheric temperature.	Kelvin

3.7 *AutoMaxTemperature Property*

3 You can use the **AutoMaxTemperature** property to get the maximum temperature of the image, most suitable for auto adjustment of the scale

Syntax

temp = *object*. **AutoMaxTemperature**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>temp</i>	The maximum existing temperature of the image when a small amount of the hottest pixels have been discarded	The temperature is in the current unit.

NOTE: You can't expect the temperature to be correct until the ThermaCAM™ Researcher has updated itself completely to the image.

SEE ALSO: For more information, see section:

- 3.38 – ImageMaxTemperature Property on page 19

3.8 *AutoMinTemperature Property*

You can use the **AutoMinTemperature** property to get the minimum temperature of the image, most suitable for auto adjustment of the scale

Syntax

temp = *object*. **AutoMinTemperature**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>temp</i>	The minimum existing temperature of the image when a small amount of the coldest pixels have been discarded	The temperature is in the current unit.

NOTE: You can't expect the temperature to be correct until the ThermaCAM™ Researcher has updated itself completely to the image.

SEE ALSO: For more information, see section:

- 3.39 – ImageMinTemperature Property on page 19

3.9 *BBRange Property*

Sets or gets the measurement range

Syntax

object.**BBRange** = *range*

range = *object*.**BBRange**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>range</i>	Integer. Measurement range.	0-N. N depends on the camera.

3.10 *CalcWhileMovingAna Property*

Set this property to FALSE to prevent the ThermaCAM™ Researcher from making analyses while some analysis function is being moved.

Syntax

object. **CalcWhileMovingAna** = *flag*

flag = *object*. **CalcWhileMovingAna**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE = Calculate while moving analysis. FALSE = Do not calculate.

NOTE: ThermaCAM™ Researcher will calculate the analysis automatically when the movement is stopped.

3.11 *Camera Property*

Gets the camera type value. This property is read only.

Syntax

*cameratyp*e = *object*.**Camera**

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>cameratype</i>	16 bit integer. Camera type	0 = No camera connected 1 = AGEMA 500 camera 2 = Thermovison 900 camera 3 = THV 1000 4 = SC1000 5 = S-series 6 = A-series

3.12 CameraFilter Property

Retrieves the type of filter used when the IR image was taken. The filter is returned as a string.

Syntax

camerafilter = *object*.**CameraFilter**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>camerafilter</i>	Camera filter	

3.13 CameraLens Property

Retrieves the type of lens used when the IR image was taken. The lens is returned as a string.

Syntax

cameralens = *object*.**CameraLens**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>cameralens</i>	Camera lens	

3.14 CameraLevel Property

Retrieves the camera level when the IR image was taken. The level value is returned as a string.

Syntax

cameralevel = *object.CameraLevel*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>cameralevel</i>	Camera level	

3

3.15 *CameraRangeMax Property*

Retrieves the max camera range when the IR image was taken. The value is returned as a string.

Syntax

camerarangemax = *object.CameraRangeMax*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>camera rangemax</i>	Camera range max	

3.16 *CameraRangeMin Property*

Retrieves the min camera range when the IR image was taken. The value is returned as a string.

Syntax

camerarangemin = *object.CameraRangeMin*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>camera rangemin</i>	Camera range max	

3.17 *CameraSerialNo Property*

Retrieves the serial number of camera used when the IR image was taken. The serial number is returned as a string.

Syntax

cameraserialno = *object.CameraSerialNo*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>camera serialno</i>	Camera serial number	

3.18 CameraSpan Property

Retrieves the camera span when the IR image was taken. The span value is returned as a string.

Syntax

cameraspan = *object*.**CameraSpan**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>cameraspan</i>	Camera span	

3.19 CameraType Property

Retrieves the type of camera used when the IR image was taken. The camera type is returned as a string.

Syntax

cameratype = *object*.**CameraType**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Cameratype</i>	String representing the camera type	

3.20 CanHaveBarCodes Property

Returns the number of bar codes possible for this type of IR image, not the number of codes actually present.

Syntax

number = *object*.**CanHaveBarCodes**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>number</i>	short integer	for example 5, for Prism DS images

3.21 *Cassette Property*

Sets or gets the camera filter cassette. This property only affects the Thermovision 900 camera.

Syntax

object.Cassette = *value*

value = *object.Cassette*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>value</i>	16-bit integer. Filter cassette number.	0-3

3.22 *ComPort Property*

Gets or sets the Comport number used to communicate with the camera.

Syntax

shNumber = *object.ComPort*

object.ComPort = *shNumber*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>shNumber</i>	Short integer.	0 = Automatic detection 1 = COM1 etc

3.23 *Connected Property*

Returns camera connection state .

Syntax

flag = *object.Connected*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Flag</i>	Boolean:Connection state	TRUE = Camera connected FALSE = Camera disconnected

3.24 Cooler Property

Sets or gets the cooler state.

Syntax

object.Cooler = *flag*

flag = *object.Cooler*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean. Cooler state.	TRUE = Cooler on. FALSE = Cooler off.

3.25 CurrentIROutput Property

Gets the current type of output: IR or other units.

Syntax

type = *object.CurrentIROutput*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>type</i>	Type of output	0 = Temperature 2 = Difference temperature 4 = Object Signal 5 = Difference Object Signal Difference unit images are created by the subtraction function

NOTE: You cannot set this property directly.

SEE ALSO: For more information, see section:

- 3.52 – PreferredIROutput Property on page 24

3.26 *DefaultHistogramClasses Property*

Sets or gets the current default number of classes for new analysis Areas or Lines.

Syntax

object.DefaultHistogramClasses = *number*

number = *object.DefaultHistogramClasses*

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>number</i>	Number of classes	Min 2, max 64.

3

3.27 *Device Property*

Gets or sets the Device (camera interface) type used.

Syntax

shDevTyper = *object.Device*

object.Device = *shDevType*

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>shDevType</i>	Short integer.	0 = None 1 = PC Card Interface 2 = IC-2 DIG 16 Frame Grabber 3 = FireWire 4-6 = Reserved 7 = IRFlashLink Frame Grabber

3.28 *DistanceUnit Property*

Sets or gets the current distance unit.

Syntax

object.DistanceUnit = *distanceunit*

distanceunit = *object.DistanceUnit*

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>distanceunit</i>	Distance unit (short integer)	0 Meter 1 Foot

NOTE: Changes will affect not only the distance in the settings dialog, but also the corresponding object parameter read-out. No event is fired by this change.

3.29 *Emissivity Property*

Sets or gets the image scale emissivity factor

Syntax

object.**Emissivity** = *factor*

factor = *object*.**Emissivity**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>factor</i>	Float. Emissivity factor.	0.01 - 1.00

3.30 *EstimatedTransmission Property*

Sets or gets the analysis estimated transmission. A value of zero, which is the default, means that the analysis system calculates its own transmission factor.

Syntax

object.**EstimatedTransmission** = *factor*

factor = *object*.**EstimatedTransmission**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>factor</i>	Float. Estimated transmission.	0.0 - 1.0, where 0.0 disables the estimation.

3.31 *ExtOpticsTemperature Property*

Sets or gets the temperature of some external optics, such as a heat shield.

Syntax

object.**ExtOpticsTemperature** = *value*

value = *object*.**ExtOpticsTemperature**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Value</i>	Float.	Kelvin temperature

3.32 *ExtOpticsTransmission Property*

Sets or gets the transmission of some external optics, such as a heat shield.

Syntax

object.**ExtOpticsTransmission** = *value*

value = *object*.**ExtOpticsTransmission**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Value</i>	Float.	0.0 to 1.0 Use 1.0 if not external optics is exists.

3.33 *Filter Property*

Sets or gets the camera filter. This property only affects the Thermovision 900 camera.

Syntax

object.**Filter** = *value*

value = *object*.**Filter**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>value</i>	16-bit integer. Filter number.	0-5

3.34 *Filtering Property*

Sets or gets the image noise filtering value. This property only affects the AGEMA 550 camera.

Syntax

object.**Filtering** = *value*

value = *object*.**Filtering**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>value</i>	16-bit integer. Noise filtering value.	0 = Normal (medium) image filtering 1 = Heavy image filtering

3

3.35 *Freeze Property*

Sets or gets the image freeze state.

Syntax

object.**Freeze** = *flag*

flag = *object*.**Freeze**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Flag indicating the image freeze state.	TRUE = Image will be frozen. FALSE = Unfreeze image

3.36 *ImageDate Property*

Retrieves the IR image date.

Syntax

imagedate = *object*.**ImageDate**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>imagedate</i>	Image date as a string.	

3.37 *ImageFilename Property*

Retrieves the IR image filename.

Syntax

imagefilename = *object*.**ImageFilename**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>image filename</i>	Image filename.	

3.38 *ImageMaxTemperature Property*

The maximum existing temperature of the IR image.

Syntax

temperature = *object*.**ImageMaxTemperature**(*tempstate*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>TempState</i>	Passed by reference. The state of the temperature will be returned here.	0 OK 1 Warning, the temperature is unreliable 2 Underflow, too low temperature 3 Overflow, too high temperature 4 Invalid, the temperature could not be calculated
<i>temperature</i>	The maximum existing temperature of the image	The temperature is in the current unit.

NOTE: You can't expect the temperature to be correct until the ThermoCAM™ Researcher has updated itself completely to the image.

3.39 *ImageMinTemperature Property*

Minimum existing temperature of the IR image.

Syntax

temperature = *object*.**ImageMinTemperature**(*tempstate*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>TempState</i>	Passed by reference. The state of the temperature will be returned here.	0 OK 1 Warning, the temperature is unreliable 2 Underflow, too low temperature 3 Overflow, too high temperature 4 Invalid, the temperature could not be calculated
<i>temperature</i>	The minimum existing temperature of the image	The temperature is in the current unit.

NOTE: You can't expect the temperature to be correct until the ThermoCAM™ Researcher has updated itself completely to the image.

3.40 *ImagePath Property*

Gets or sets the image directory.

3

Syntax

str = **object.ImagePath**

object.ImagePath = *str*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>str</i>	String.	"C:\Images" for example

3.41 *ImageTime Property*

Retrieves the IR image time in seconds since January 1 1970.

Syntax

longTime = **object.ImageTime**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>longTime</i>	Time in seconds (long integer)	

SEE ALSO: If you are interested in a millisecond resolution, see section:

- 3.42 – ImageTimeMSec Property on page 20

3.42 *ImageTimeMSec Property*

Retrieves the IR image time in seconds since January 1 1970. The number of milliseconds within that second is returned in the fraction part.

Syntax

time = **object.ImageTimeMSec**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>Time</i>	Image time in seconds (integer part), number of milliseconds in the fraction part	

3.43 *ImageTimeStr* Property

Retrieves the IR image time as a string.

Syntax

imagetime = *object*.**ImageTimeStr**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>imagetime</i>	Image time as a string.	

3.44 *ImageTitle* Property

Retrieves the IR image title.

Syntax

imagetitle = *object*.**ImageTitle**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>imagetitle</i>	Image title.	

3.45 *IsFirstImage* Property

This property is TRUE when the first image in the session is displayed.

Syntax

first = *object*.**IsFirstImage**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>first</i>	First image	TRUE or FALSE

3.46 *IsLastImage Property*

This property is TRUE when the last image in the session is displayed.

Syntax

last = *object*.**IsLastImage**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>last</i>	Last image	TRUE or FALSE

3.47 *IsUpdated Property*

Reveals if ThermoCAM™ Researcher has been updated to the current image or not.

Syntax

flag = *object*.**IsUpdated**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	

SEE ALSO: To get the update done immediately, see section:

- 4.57 – Refresh Method on page 63

3.48 *KeepAnalysis Property*

The **KeepAnalysis** property controls if the current analysis shall be kept when a new IR image is loaded or not.

Syntax

object.**KeepAnalysis** = *keep*

keep = *object*.**KeepAnalysis**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>keep</i>	Boolean	TRUE = do keep the current analysis FALSE = don't

3.49 *Motor Property*

Sets or gets the image freeze state.

Syntax

object.Motor = *state*

state = *object.Motor*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>State</i>	16-bit integer. Motor/cooler state.	0 = Motor off 1 = Motor on 4 = Stirling cooler standby

3

3.50 *ObjectDistance Property*

Sets or gets the measurement object distance value.

Syntax

object.ObjectDistance = *distance*

distance = *object.ObjectDistance*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Distance</i>	Float. Object distance.	0.0 - 9999.9 meters 0.0 - 32808.1 feet

NOTE: The current distance unit affects the value range.

3.51 *OLELiveLinkDelay Property*

Sets or gets the update delay in ms for OLE links.

Syntax

object.OLELiveLinkDelay = *delay*

delay = *object.OLELiveLinkDelay*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>delay</i>	OLE link delay in milli-seconds	

3.52 PreferredIROutput Property

Sets the preferred type of output: IR or other units.

Syntax

object.PreferredIROutput = *type*

type = *object*.PreferredIROutput

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>type</i>	Type of output	0 = Temperature 1 - 3 Reserved 4 = Object Signal 5 - 10 Reserved

NOTE: This property states the desired output of ThermoCAM™ Researcher. The program will choose an output that corresponds as closely to this setting as possible.

SEE ALSO: For more information about current output, see section:

- 3.25 – CurrentIROutput Property on page 14

3.53 Ratio Property

Gets the current image aspect ratio.

Syntax

doubleRatio = *object*.Ratio

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>doubleRatio</i>	The current aspect ratio in double precision.	The image width divided by its height.

3.54 RecAutoNameBase Property

Set or get user defined prefix for image names when recording. For each new image recorded Researcher will add an incremented sequence number.

Syntax

prefix = *object*.RecAutoNameBase

object.RecAutoNameBase = *prefix*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>prefix</i>	String: Image prefix	For example "MyImage"

3.55 *ReclmgPres Property*

Gets or sets the Enable Image Presentation option during image recording

Syntax

bool = *object*.**ReclmgPres**

object.**ReclmgPres** = *bool*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>bool</i>	Boolean.	

3.56 *RecOneFileRec Property*

Set or get whether recorded images should be stored in separate files or in one sequence file.

Syntax

flag = *object*.**RecOneFileRec**

object.**RecOneFileRec** = *flag*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Flag</i>	Boolean: Storing condition	TRUE = Store images in one sequence file FALSE = Store images in separate files

3.57 *RecStartCond Property*

Set or get start condition for recording of images. Recording may start on four different events:

- User selects the start/stop button on the recording toolbar.
- User pushes the function key F5.
- Predefined time.
- External trig event

Syntax

index = *object*.**RecStartCond**

object. **RecStartCond** = *index*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Index</i>	Short: Start condition	0 = Tool button/Key trig 1 = Time 2 = External trig+delay

3.58 *RecStartDelay Property*

Set or get delay time for start of recording.

Syntax

delay = *object*.**RecStartCond**

object. **RecStartCond** = *delay*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Delay</i>	Double: delay time	

NOTE: This value is only valid if start condition is set to 3 (External Trig).

3.59 *RecStartTime Property*

Set or get time for recording to start.

Syntax

time = *object*.**RecStartCond**

object. **RecStartCond** = *time*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Time</i>	Double: starting time	

NOTE: This value is only valid if start condition is set to 2 (Time).

3.60 *RecStopCond Property*

Set or get condition for stop recording. Recording may stop on five different events:

- User selects the start/stop button on the recording toolbar.
- After XX images.
- Predefined time.
- User pushes the function key F5.
- External trig event

3

Syntax

index = *object*.**RecStopCond**

object.**RecStopCond** = *index*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Index</i>	Short: Stop condition	0 = Tool button/Key trig 1 = Time 2 = After N images 3 = External trig+delay

3.61 *RecStopDelay Property*

Set or get delay time for stop recording.

Syntax

delay = *object*.**RecStopDelay**

object.**RecStopDelay** = *delay*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Delay</i>	Double: Delay time	

NOTE: This value is only valid if stop condition is set to 4 (External trig).

3.62 *RecStopDuration Property*

Set or get recording time duration.

Syntax

time = *object*.**RecStopDuration**

object.**RecStopDuration** = *time*

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>Time</i>	Double: Recording time	

NOTE: This value is only valid if stop condition is set to 2 (Time).

3.63 *RecStopImages Property*

Set or get number of images to record.

Syntax

nolimages = *object*.**RecStopImages**

object.**RecStopImages** = *nolimages*

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>nolimages</i>	Short: Number of images to record	

NOTE: This value is only valid if stop condition is set to 1 (Stop after XX images).

3.64 *RecStoreCond Property*

Set or get condition for storing images during recording. An image may stored under five different conditions:

- Highest possible speed.
- At a Time interval.
- At each N:th image
- User pushes the function key F5.
- External trig event

Syntax

index = *object*.**RecStoreCond**

object.**RecStoreCond** = *index*

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>Index</i>	Short: Storing condition	0 = Tool button/Key trig 1 = Highest speed 2 = Every N images 3 = Time interval 4 = External trig+delay

3

3.65 *RecStoreImages Property*

Set or get image interval for storing images.

Syntax

$N = \text{object}.\text{RecStoreImages}$

$\text{object}.\text{RecStoreImages} = N$

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>noImages</i>	short: Store every N:th image	

NOTE: This value is only valid if storing condition is set to 2 (image interval).

3.66 *RecStoreInterval Property*

Set or get time interval for storing images.

Syntax

$\text{time} = \text{object}.\text{RecStoreInterval}$

$\text{object}.\text{RecStoreInterval} = \text{time}$

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>time</i>	double: time elapsed between each stored image	

NOTE: This value is only valid if storing condition is set to 1 (Time interval).

3.67 *ReferenceTemperature Property*

Sets or gets the analysis reference temperature.

Syntax

object.**ReferenceTemperature** = *temperature*

temperature = *object*.**ReferenceTemperature**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>temperature</i>	Float. Reference temperature in Kelvin.	0.0 - 9999.9

3.68 *RefTemperatureExists Property*

Indicates if the IR image reference temperature exists

Syntax

bFlag = *object*.**RefTemperatureExists**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>bFlag</i>	Boolean	TRUE if a reference temperature exist otherwise FALSE.

3.69 *RelativeHumidity Property*

Set or get the relative humidity value.

Syntax

object.**RelativeHumidity** = *factor*

factor = *object*.**RelativeHumidity**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Factor</i>	Float. Relative humidity factor.	0.00 - 0.99

3.70 *Save900Analysis Property*

Sets or gets the flag indicating if analysis shall be stored when saving images with Thermovision 900 and Thermovision 1000 format or not.

Syntax

object.**Save900Analysis** = *flag*

flag = *object*.**Save900Analysis**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE = do save the analysis in the image FALSE = don't

NOTE: The image format obtained when the analysis is saved in the image is not compatible with the normal Thermovision 900 image format. Such new images are not recognised by the Thermovision 900 system controller.

3

3.71 *ShowAnalysis Property*

Set or get the flag indicating if analysis symbols shall be displayed at all on the IR image.

Syntax

object.**ShowAnalysis** = *flag*

flag = *object*.**ShowAnalysis**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE Display the analysis symbols FALSE Don't.

NOTE: This property does not apply to the flying spot function.

3.72 *ShowAnalysisBackColor Property*

Set or get the flag indicating if the analysis symbol label shall be drawn transparently or not.

Syntax

object.**ShowAnalysisBackColor** = *flag*

flag = *object*.**ShowAnalysisBackColor**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Keep</i>	Boolean	TRUE Analysis symbol labels are not transparent FALSE They are.

NOTE: Changes to this property it will only affect new analysis symbol labels.

3.73 *ShowAnalysisTemp Property*

Sets or gets the flag indicating if a temperature shall be displayed along with the analysis symbol name.

3

Syntax

object.**ShowAnalysisTemp** = *flag*

flag = *object*.**ShowAnalysisTemp**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE A temperature will be displayed FALSE A temperature will not be displayed.

NOTE: Changes to this property it will only affect new analysis symbol labels. It is not relevant for the flying spot function.

3.74 *ShowIlegTempIndicator Property*

By this property you can control whether an indication shall be made when a temperature is unreliable or not. It becomes unreliable when it is close to or outside the calibration of the camera.

The indicators are:

*	Warning, temperature unreliable
<	Temperature underflow
>	Temperature overflow

You are most likely to use this property when you link values by OLE to spread sheet programs which do not want to calculate on such "decorated" digits.

Syntax

object.**ShowIlegTempIndicator** = *flag*

flag = *object*.**ShowIlegTempIndicator**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>flag</i>	Boolean	TRUE = do indicate unreliable results FALSE = don't

3.75 *ShowLabels Property*

This property will control if the labels of the analysis functions are shown along with the symbols on top of the image or not. The appearance of analysis symbols as such is controlled by the **ShowAnalysis Property**

Syntax

object.**ShowLabels** = *flag*

flag = *object*.**ShowLabels**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE = Do show the labels. FALSE = Don't.

SEE ALSO: The presence of a measurement result within the label is controlled by the following property:

- 3.73 – ShowAnalysisTemp Property on page 32

3.76 *ShowTemperatureScale Property*

Show or hide the temperature scale.

Syntax

object.**ShowTemperatureScale** = *show*

show = *object*.**ShowTemperatureScale**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>show</i>	Boolean	TRUE Show the temperature scale FALSE Don't.

3.77 *TargetMotion Property*

Sets or gets the target motion value. This property only affects the AGEMA 550 camera.

Syntax

object.TargetMotion = *value*

value = *object.TargetMotion*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>value</i>	Integer. Target motion value.	0 = Still or slow moving objects 1 = Fast moving objects

3.78 *TemperatureUnit Property*

Sets or gets the temperature unit.

Syntax

object.TemperatureUnit = *temperatureunit*

temperatureunit = *object.TemperatureUnit*

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Temperature-unit</i>	Temperature unit (short integer)	0 Celsius 1 Fahrenheit 2 Kelvin

NOTE: No event is fired by this change.

3.79 *TotalImages Property*

Get the number of selected images for playing.

Syntax

images = *object.TotalImages*

object.TotalImages = *images*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Images</i>	Selected number of images	

3.80 *UseImageObjParams Property*

Set or get the state that tells if current object parameters shall be kept when a new image is loaded or if they shall be taken from the new image.

Syntax

object.UseImageObjParams = *usage*

usage = *object.UseImageObjParams*

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>usage</i>	Boolean	TRUE Take the object parameters from the new image, when it is loaded. FALSE Keep the current object parameters when a new image is loaded.

3.81 *UseImageScale Property*

Set or get the state that tells if current scale shall be kept when a new image is loaded or if it shall be taken from the new image.

Syntax

object.UseImageScale = *usage*

usage = *object.UseImageScale*

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>usage</i>	Boolean	TRUE Take the scale from the new image, when it is loaded. FALSE Keep the current scale when a new image is loaded.

3.82 *UseImageZoom Property*

The **UseImageZoom** property controls if the current zoom factor shall be kept or not when a new IR image is loaded.

Syntax

object.UseImageZoom = *keep*

flag = *object.UseImageZoom*

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE = keep current zoom factor FALSE = don't keep current zoom factor

3.83 *Version Property*

Get the current version string.

Syntax

string = *object*.**Version**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>string</i>	Version	"ThermaCAM™ Researcher Version x.xx"

3.84 *View3D Property*

Set or get the flag that tells if the IR image shall be displayed in 3D or 2D.

Syntax

object.**View3D** = *flag*

flag = *object*.**View3D**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE Show IR image in 3D FALSE Show IR image in 2D

NOTE: 3D view will decrease the performance of show images quit dramatically. It is also not possible to display any analysis in the 3D view.

3.85 *Visible Property*

Get or set the visibility of the Researcher window.

Syntax

object.**Visible** = *flag*

flag = *object*.**Visible**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE = Researcher is visible FALSE = Researcher is hidden

3.86 Zoomfactor Property

Sets or gets the IR image zoom factor.

Syntax

object.**Zoomfactor** = *zoomfactor*

zoomfactor = *object*.**Zoomfactor**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>zoomfactor</i>	Double precision	>= 1.0, <= 4.0

NOTE: The zooming is performed around the centre of the IR image. Changes to this property do not fire any event.

INTENTIONALLY LEFT BLANK

4 Methods

4.1 *AddPlaybTextCmt Method*

Adds (or changes) text comments during image playback.

Syntax

Result = *object*.**AddPlaybTextCmt**(*strLabel*,*strValue*,*bChgExisting*)

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>strLabel</i>	The name of the text comment to be added.	"Label1", for example
<i>strValue</i>	The value as a string name of the text comment to be added.	"Value1", for example
<i>bChgExisting</i>	Boolean indicating if the Label may exist in advance.	If bChgExisting is TRUE, and the label already exists, its value string is changed.
<i>Result</i>	Boolean	TRUE: Successful outcome.

NOTE: This method can only be used with image formats that can contain text comments, and not during the display of live images.

4.2 *AddRecTextCmt Method*

Adds (or changes) text comments during image recording.

Syntax

Result = *object*.**AddRecTextCmt**(*strLabel*,*strValue*,*bChgExisting*)

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>strLabel</i>	The name of the text comment to be added.	"Label1", for example
<i>strValue</i>	The value as a string name of the text comment to be added.	"Value1", for example
<i>bChgExisting</i>	Boolean indicating if the Label may exist in advance.	If bChgExisting is TRUE, and the label already exists, its value string is changed.

Item	Description	Value
<i>Result</i>	Boolean	TRUE: Successful outcome.

NOTE: This method can only be used with SC 2000, SC 3000, S-series and A-Series cameras, and only while the camera is connected. These text comments are included in the recorded images.

4.3 *AnaDefineFormula Method*

(Re)define a formula for the active session. You may define all formula parameters yourself, or let ThermoCAM™ Researcher assist you by automatically assigning formula name or presenting a formula editing dialog.

Syntax

FormulaName = *object*.**AnaDefineFormula**(*Name*,*Label*,*Expression*, *Precision*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>Name</i>	The name of the formula to be (re)-defined. Enter an empty string to make ThermoCAM™ Researcher pick a free name.	"fo01", "fo02", "fo03",... or ""
<i>Label</i>	User friendly description of formula	For example: SpotAreaSubtraction
<i>Expression</i>	Formula expression. Enter an empty string to show formula editing dialog.	For example: {sp1.temp} – {ar1.avg}
<i>Precision</i>	Short: Number of decimals in result.	0 - 5
<i>FormulaName</i>	If formula definition was successful, the name of the formula will be returned.	Ok: "fo01", "fo02", "fo03",... Precision < 0: "Precision must be >= 0" Error in formula: "FormulaError"

4.4 *AnaDeleteFormula Method*

Delete formula(s) for the active session.

Syntax

Result = *object*.**AnaDeleteFormula**(*Name*)

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>Name</i>	The name of the formula to be deleted. "*" to make ThermaCAM™ Researcher delete all formulas.	"fo01" or "*"
<i>Result</i>	Boolean	TRUE: Successful request.

4

4.5 *AnalysisDefineBentLine Method*

The **AnaDefineBentLine** method lets you add, move or reshape a linemeasurement function on the IR Image. This type of line can have many corners.

Syntax

usedFuncName = *object*.**AnalysisDefineBentLine**(*useFuncName*, *numPointss*, *sh-PointArray*)

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>UseFuncName</i>	The name of the line to be (re)-defined. Enter an empty string to make ThermaCAM™ Researcher pick a free name.	"li1" or ""
<i>NumPoints</i>	Integer indicating the number of corners present in the integer array	For example: 3
<i>ShPointArray</i>	Short integer array filled with the coordinates of the corners of the line, the beginning and end points included. The size is 2*numPoints. The coordinates come in pairs with the xpositions before the ypositions	For example: 10,20 30,40 5,35
<i>UsedFuncName</i>	The name of the function that was (re)defined.	

SEE ALSO: For information about straight lines, see section:

- 4.8 – AnalysisDefineLine Method on page 43

NOTE: You can't expect any results from this function until the ThermoCAM™ Researcher has updated itself completely to the new conditions.

4.6 AnalysisDefineBoxArea Method

The **AnalysisDefineBoxArea** method lets you add, move or reshape a box shaped area measurement function on the IR Image.

4

Syntax

usedFuncName = *object*.**AnalysisDefineBoxArea**(*useFuncName*, *xPos*, *yPos*, *width*, *height*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>useFuncName</i>	The name of the area to be (re)-defined. Enter an empty string to make ThermoCAM™ Researcher pick a free name.	"ar1" or ""
<i>xPos,yPos</i>	Integer coordinates within the current image of the top left corner of the box.	10,10, for example. 0,0 corresponds to the top left corner of the image.
<i>width,height</i>	Integer width and height of the box.	
<i>usedFuncName</i>	The name of the function that was (re)defined.	

NOTE: You can't expect any results from this function until ThermoCAM™ Researcher has updated itself completely to the new conditions.

4.7 AnalysisDefineCircleArea Method

The **AnalysisDefineCircleArea** method lets you add, move or reshape a circle shaped area measurement function on the IR Image.

Syntax

usedFuncName = *object*.**AnalysisDefineCircleArea**(*useFuncName*, *xCenter*, *yCenter*, *radius*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-researcher object	For example Session1.
<i>useFuncName</i>	The name of the area to be (re)-defined. Enter an empty string to make ThermaCAM™ Researcher pick a free name.	"ar1" or ""
<i>xCenter, yCenter</i>	Integer coordinates within the current image of the top left corner of the box.	10,10, for example. 0,0 corresponds to the top left corner of the image.
<i>radius</i>	Integer radius of the circle.	
<i>usedFuncName</i>	The name of the function that was (re)defined.	

NOTE: You can't expect any results from this function until ThermaCAM™ Researcher has updated itself completely to the new conditions.

4.8 AnalysisDefineLine Method

The **AnalysisDefineLine** method lets you add, move or reshape a linemeasurement function on the IR Image.

Syntax

usedFuncName = *object*.**AnalysisDefineLine**(*useFuncName*, *xFirst*, *yFirst*, *xLast*, *yLast*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-researcher object	For example Session1.
<i>useFuncName</i>	The name of the spotmeter to be (re)-defined. Enter an empty string to make ThermaCAM™ Researcher pick a free name.	"li1" or ""
<i>xFirst, yFirst, xLast, yLast</i>	Short integer coordinates of the line.	10,10 50,60, for example. 0,0 corresponds to the top left corner of the image.
<i>usedFuncName</i>	The name of the function that was (re)defined.	

SEE ALSO: For information about straight lines, see section:

- 4.5 – AnalysisDefineBentLine Method on page 41

NOTE: You can't expect any results from this function until the ThermaCAM™ Researcher has updated itself completely to the new conditions.

4.9 AnalysisDefinePolygonArea Method

The **AnalysisDefinePolygonArea** method lets you add, move or reshape a polygon shaped area measurement function on the IR Image.

4

Syntax

usedFuncName = *object*.**AnalysisDefinePolygonArea**(*useFuncName*, *numCorners*, *shCornerArray*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-researcher object	For example Session1.
<i>useFuncName</i>	The name of the area to be (re)-defined. Enter an empty string to make ThermaCAM™ Researcher pick a free name.	"ar1" or ""
<i>numCorners</i>	Integer indicating the number of corners present in the integer array	For example: 3
<i>shCornerArray</i>	Short integer array filled with the coordinates of the corners of the area. The size is 2*numCorners. The coordinates come in pairs with the xpositions before the ypositions	For example: 10,20 30,40 5,35
<i>radius</i>	Integer radius of the circle.	
<i>usedFuncName</i>	The name of the function that was (re)defined.	

NOTE: You can't expect any results from this function until ThermaCAM™ Researcher has updated itself completely to the new conditions.

4.10 AnalysisDefineSpotmeter Method

The **AnalysisDefineSpotmeter** method lets you add or move a spotmeter on the IR Image.

Syntax

usedFuncName = *object*.**AnalysisDefineSpotmeter**(*useFuncName*, *xPos*, *yPos*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>useFuncName</i>	The name of the spotmeter to be (re)-defined. Enter an empty string to make ThermoCAM™ Researcher pick a free name.	"sp1" or ""
<i>xPos,yPos</i>	Integer coordinates within the current image.	10,10, for example. 0,0 corresponds to the top left corner of the image.
<i>usedFuncName</i>	The name of the function that was (re)defined.	

NOTE: You can't expect any results from this function until ThermoCAM™ Researcher has updated itself completely to the new conditions.

4.11 *AnalysisDeleteFunc Method*

The **AnalysisDeleteFunc** method lets you remove a spotmeter, area, line or isotherm from the IR Image.

Syntax

flag = *object*.**AnalysisDeleteFunc**(*funcName*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>funcName</i>	The name of the function to be removed. By indicating "*" you can remove all functions at once.	For example "sp1"
<i>flag</i>	Boolean	TRUE if the function has been removed

4.12 *AnalysisFuncNames Method*

You can use the **AnalysisFuncNames** method to become informed about the currently active analysis functions (spotmeters, areas, lines and isotherms).

Syntax

string = *object*.**AnalysisFuncNames**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>string</i>	Active function names separated by line feeds.	For example: sp01 ar03 li02 iso01

NOTE: These names will be the generic ones even if you redefine the labels that are displayed on the screen. You can not expect the names to be sorted.

4.13 AnalysisHistoArray Method

The **AnalysisHistoArray** method retrieves the current histogram of an area or line. Set the number of classes by the **AnalysisHistoClasses Method**. Set the histogram limits by the **SetScaling Method**.

Syntax

number = *object*.**AnalysisHistoArray**(*anaFuncName*, *dblLowLim*, *dblHighLimTemp*, *dblUflPercent*, *dblOflPercent*, *fltPercentArray*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>anaFuncname</i>	Generic name string of an area or line analysis function	"AR01"
<i>dblLowLim</i> , <i>dblHighLim</i>	double temperature values in which the lower and upper limits of the histogram are returned, if successful.	The current presentation unit is used.
<i>dblUflPercent</i>	Percentage of the temperatures that are below lower limit	Percentage from 0.0 to 100.0.
<i>dblOflPercent</i>	Percentage of the temperatures that are above upper limit	Percentage from 0.0 to 100.0.
<i>fltPercentArray</i>	Float array that will be filled with the corresponding histogram in ascending order.	Percentages from 0.0 to 100.0.
<i>number</i>	Number of classes being returned. Positive if successful	

NOTE: The *Number* returned in case of failure is the negative value of the number of classes.

NOTE: You can't expect the array to be correct until ThermaCAM™ Researcher has updated itself completely to the image.

NOTE: The width of each histogram class equals $(dblHighLim - dblLowLim)/number$.

4.14 *AnalysisHistoClasses Method*

The **AnalysisHistoClasses** method will permit you to get and set the number of classes used when the histogram is calculated for an area or line. Use the **AnalysisHistoArray Method** to retrieve the actual histogram.

Syntax

object.**AnalysisHistoClasses**(*anaFuncName*) = *number*

number = *object*.**AnalysisHistoClasses**(*anaFuncName*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>anaFuncname</i>	Generic name string of an area or line analysis function	AR01
<i>number</i>	Number of classes to be used	

NOTE: You can't expect the temperatures to be correct until ThermaCAM™ Researcher has updated itself completely to the new conditions.

4.15 *AnalysisHistoThreshold Method*

The **AnalysisHistoThreshold** method lets you retrieve and change the threshold temperature of the specified area and line measurement functions of an IR image. Note that this method only applies to areas and lines.

Syntax

object.**AnalysisHistoThreshold**(*funcName*) = *temperature*

temperature = *object*.**AnalysisHistoThreshold**(*funcName*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>funcName</i>	The name of the function to be affected.	"ar1", for example

Item	Description	Value
<i>temperature</i>	Double: The temperature at which the histogram threshold shall be set	The current presentation unit is used.

NOTE: You can't expect the temperatures to be correct until ThermaCAM™ Researcher has updated itself completely to the new conditions.

4.16 *AnalysisResults Method*

Use the **AnalysisResults** method to retrieve some or all analysis single results as strings.

Syntax

resultString = *object*.**AnalysisResults**(*anaFuncSelString*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>anaFuncSelString</i>	A string that identifies the selection desired.	"sp5" Only spotmeter 5 is wanted "sp*" All spotmeters are wanted. "ar1.avg" Only area 1 average wanted. "ar*.avg" All area averages wanted (similarly for lines, areas, isotherms) "*" Everything is wanted.
<i>resultString</i>	A list with the desired results. One name and value on each line. The name and the value are separated by a TAB character (CHR\$(9))	For example: sp01.label <tab> SP01 <linefeed> sp01.temp <tab> 35.22 <linefeed> sp01.diff <tab> 0.22 <linefeed> sp01.emis <tab> ILLEGAL <linefeed> and so on.

NOTE: You will **not** receive any multiple results, such as the line stringdata and histograms, in this way.

NOTE: You can expect the string ILLEGAL to appear as a value if the value is undefined.

NOTE: You can't expect the results to be correct until ThermaCAM™ Researcher has updated itself completely to the image.

SEE ALSO: For information about performing an immediate update, see section:

- 4.57 – Refresh Method on page 63

4.17 *AnalysisTemperatureArray Method*

You can use the **AnalysisTemperatureArray** method to retrieve a whole set of temperatures for a line or a box area.

Syntax

Number = *object*.**AnalysisTemperatureArray**(*funcName*,*fltTempArray*, *sStateArray*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.
<i>funcname</i>	Generic name string of a line or box area analysis function	LI01
<i>fltTempArray</i>	Float array that will be filled with the temperature values.	The current presentation unit is used. Box area temperatures are stored row by row.
<i>sStateArray</i>	Array of short integers that will be filled with the corresponding temperature state values	0 OK 1 Warning, the temperature is unreliable 2 Underflow, too low temperature 3 Overflow, too high temperature 4 Invalid, the temperature could not be calculated.
<i>Number</i>	The number of elements stored in the array. Long integer. Positive if successful.	

NOTE: The *Number* returned in case of failure is the negative value of the number required for this line or area.

NOTE: You can't expect the array to be correct until ThermaCAM™ Researcher has updated itself completely to the image.

4.18 *AnalysisTemperatureArrayLen Method*

You can use the **AnalysisTemperatureArrayLen** method to retrieve the number of array elements required for a call to the **AnalysisHistoArray Method** or the **AnalysisTemperatureArray Method**

Syntax

Number = *object*.**AnalysisTemperatureArrayLen**(*lineFuncname*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>lineFuncname</i>	Generic name string of a line analysis function	LI01
<i>Number</i>	The minimum number of elements required for the array	

4

4.19 *AnalysisTemperatureByName Method*

You can use the **AnalysisTemperatureByName** method to retrieve one temperature value for an analysis function.

Syntax

flag = *object*.**AnalysisTemperatureByName**(*strGenericName*,*dblTemperature*,*sTempState*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>strGenericName</i>	Name of the analysis function plus the type of temperature requested separated with a dot.	For example: sp1.temp ar2.avg
<i>dblTemperature</i>	The temperature for the specified analysis function. If <i>TempState</i> is Invalid, the temperature can't be determined	The current presentation unit is used.
<i>TempState</i>	The state of the temperature will be returned here.	0 OK 1 Warning, the temperature is unreliable 2 Underflow, too low temperature 3 Overflow, too high temperature 4 Invalid, the temperature could not be calculated
<i>flag</i>	Indicates if the specified name exists or not.	Boolean

NOTE: Only single temperature values can be retrieved. No emissivity, position or other information is available through this method.

NOTE: You can only retrieve temperature values, not set the values. If the analysis function does not exist, the return flag will be FALSE and *TempState* will be invalid.

NOTE: You can't expect the temperature to be correct until ThermoCAM™ Researcher has updated itself completely to the image.

4.20 *AutoAdjust Method*

Use the **AutoAdjust** method to automatically adjust the scale and colors of the displayed image.

Syntax

object.AutoAdjust

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

4.21 *AutoRewind Method*

Use the **AutoRewind** method to set the auto-rewind flag in playing disk image sequences. If set, the image sequence will automatically rewind itself when reaching the last or the first image in the selected sequence and continue to play.

Syntax

object.AutoRewind(*flag*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>flag</i>	Boolean	TRUE to enable auto-rewind

4.22 *Connect Method*

Use the **Connect** method to connect the application to the camera. This method is used to reconnect the camera again after a Disconnect command.

Syntax

object.Connect

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

4.23 *CopySessionAndImage Method*

Copies the IR-image to the clipboard with active OLE-link.
No return value.

Syntax

object.**CopySessionAndImage**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.24 *Disconnect Method*

Use the **Disconnect** method to disconnect the application from the camera.

Syntax

object.**Disconnect**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.25 *ExternalCorrection Method*

Use the **ExternalCorrection** method to calibrate the pixel to pixel differences in the camera with the present image as reference.

Syntax

object.**ExternalCorrection**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.26 *FastBackward Method*

You can use the **FastBackward** method to continuously replay the selected disk image(s) backward fast.

Syntax

object.**FastBackward**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.27 *FastForward Method*

You can use the **FastForward** method to continuously replay the selected disk image(s) forward fast.

Syntax

object.FastForward

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.28 *FocusFar Method*

Use the **FocusFar** method to start the camera focus towards infinity.

Syntax

object.FocusFar

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.29 *FocusNear Method*

Use the **FocusNear** method to start the camera focus towards near focus.

Syntax

object.FocusNear

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.30 *FocusStop Method*

Use the **FocusStop** method to stop the camera focus.

Syntax

object.FocusStop

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.31 *GetFormulaByIndex Method*

Syntax

Result = *object*.**GetFormulaByIndex**(*Index*, *Name*, *Label*, *Expression*, *Precision*)

4

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>Index</i>	[in] Index for formula sought.	"1", "2", ... "N". Where N is the total number of formulas in the session
<i>Name</i>	[out] formula name	
<i>Label</i>	[out] Formula label	
<i>Expression</i>	[out] Formula expression	
<i>Precision</i>	[short, out] Formula precision	
<i>Result</i>	Boolean	TRUE if the request was successful.

4.32 *GetFormulaByName Method*

Syntax

Result = *object*.**GetFormulaByName**(*Index*, *Name*, *Label*, *Expression*, *Precision*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>Name</i>	[in] Formula name for formula sought.	
<i>Label</i>	[out] Formula label	
<i>Expression</i>	[out] Formula expression	
<i>Precision</i>	[short, out] Formula precision	
<i>Result</i>	Boolean	TRUE if the request was successful.

4.33 *GetFrameRate Method*

Despite its name, this method gets the image rate of the current connected camera.

Syntax

object.**GetFrameRate**(*dblImageRate*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>dblImageRate</i>	Current image rate as a double precision float value	Normally 50 Hz (PAL) or 60 Hz (NTSC)

NOTE: The image rate will be rounded to the nearest Hz.

NOTE: Certain camera types that have a mode in which several small images are packed into one single frame. The image rate is increased as the image height is decreased while the frame rate stays the same. However, you will still get the image rate value.

4.34 *GetNamedValue Method*

Use the **GetNamedValue** method to retrieve application values. Get the current value of a named parameter (name). The easiest way to find the name of a particular parameter is to copy it to the clipboard with “Edit->Copy value” and link it to a spread sheet program. Then study the name in the cell. Use only the last part of the link name here.

Syntax

val = *object*.**GetNamedValue**(*named_value*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>named_value</i>	String. Name of value to retrieve.	E. g. “sp1.temp”.
<i>val</i>	String. Specified value	E.g. a spot meter temperature value.

4.35 *GetNoFormulas Method*

Syntax

Result = *object*.**GetNoFormulas**()

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

Item	Description	Value
<i>Result</i>	Short: Total number of defined formulas in the session.	

4.36 *GetNumPlaybTextCmts Method*

Retrieves the current number of text comments during image playback.

Syntax

4

shNumber = *object*.**GetNumPlaybTextCmts**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>shNumber</i>	Short integer	The number of text comments.

4.37 *GetNumRecTextCmts Method*

Retrieves the current number of text comments during image recording.

Syntax

shNumber = *object*.**GetNumRecTextCmts**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>shNumber</i>	Short integer	The number of text comments.

4.38 *GetPaletteInfo Method*

You can use the **GetPaletteInfo** method to retrieve information about the current palette (=IR colours).

Syntax

string = *object*.**GetPaletteInfo**(*Invert*,*Saturation*,*OutOfRange*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

Item	Description	Value
<i>Invert Saturation Out-OfRange</i>	Information about the current state of the invert, show saturation and show out of range colours is returned here	TRUE/FALSE
<i>string</i>	The file name of the current palette is returned here	

4.39 *GetPlaybTextCmt Method*

Retrieves text comments during image playback.

Syntax

Result = *object*.**GetPlaybTextCmt**(*nIndex*,*strLabel*,*strValue*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>nIndex</i>	The text comment wanted	0,1,2....
<i>strLabel</i>	String	The retrieved name.
<i>strValue</i>	String	The retrieved value.
<i>Result</i>	Boolean	TRUE: Successful outcome.

4.40 *GetPlaybTextCmtByName Method*

Retrieves text comments during image playback.

Syntax

Result = *object*.**GetPlaybTextCmtByName**(*strLabel*,*strValue*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>strLabel</i>	String	The comment to retrieve.
<i>strValue</i>	String	The retrieved value.
<i>Result</i>	Boolean	TRUE: Successful outcome.

4.41 *GetPlayRate Method*

Get the image playback display rate. A fast forward/backward display rate is achieved by just showing every N:th image. A slow forward/backward display is achieved by an X-fold delay of the image presentation.

Syntax

nRate = *object*.**GetPlayRate**(*bDelay*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>bDelay</i>	Boolean	TRUE if the image presentation is delayed, instead of speeded up.
<i>nRate</i>	Short integer	The N (or X) value

4.42 *GetRecTextCmt Method*

Retrieves text comments during image recording.

Syntax

Result = *object*.**GetRecTextCmt**(*nIndex*,*strLabel*,*strValue*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>nIndex</i>	The text comment wanted	0,1,2....
<i>strLabel</i>	String	The retrieved name.
<i>strValue</i>	String	The retrieved value.
<i>Result</i>	Boolean	TRUE: Successful outcome.

4.43 *GetRecTextCmtByName Method*

Retrieves text comments during image recording.

Syntax

Result = *object*.**GetRecTextCmtByName**(*strLabel*,*strValue*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

Item	Description	Value
<i>strLabel</i>	String	The name to retrieve.
<i>strValue</i>	String	The retrieved value.
<i>Result</i>	Boolean	TRUE: Successful outcome.

4.44 *GetScaling Method*

You can use the **GetScaling** method to retrieve general information about the scale

Syntax

scalelimtype = *object*.**GetScaling**(*min*, *max*, *coldistrtype*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>min,max</i>	The scale minimum and maximum temperatures will be returned here in double precision	15.8 33.6 The temperature is in the current unit.
<i>coldistrtype</i>	The current type of colour distribution is returned here	0 Temperature linear 1 Enhanced contrast (histogram equalization) 2 Camera linear
<i>scalelimtype</i>	The current type of scaling	0 Limits from image 1 Autoadjusted limits 2 Fix limits

NOTE: You can't expect the temperature to be correct until ThermaCAM™ Researcher has updated itself completely to the image.

4.45 *GotoFirstImage Method*

Use the **GotoFirstImage** method to stop playing disk images and display the first image in the selected set of disk images.

Syntax

object.**GotoFirstImage**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.

4.46 *GotoLastImage Method*

Use the **GotoLastImage** method to stop playing disk images and display the last image in the selected set of disk images.

Syntax

object.**GotoLastImage**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.

4.47 *ImageDimensions Method*

You can use the **ImageDimensions** method to retrieve information about the width and height in pixels of the current IR image.

Syntax

object.**ImageDimensions**(*width,height*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>width</i>	Integer that will receive the width of the image	For example 320
<i>height</i>	Integer that will receive the height of the image	For example 240

NOTE: The dimensions of the actual data is returned, not the dimensions visible on the screen.

4.48 *InternalCorrection Method*

Use the **InternalCorrection** method to calibrate the pixel to pixel differences in the camera with the internal shutter as reference. This method only affects the AGEMA 500 camera.

Syntax

object.**InternalCorrection**

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.

4.49 *KeyTrigF5 Method*

Simulates selection of the function key, F5, in Researcher.

Syntax

object.**KeyTrigF5**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

4

4.50 *LoadImage Method*

Use the **LoadImage** method to load a particular image file from disk and display it (full path required).

Syntax

object.**LoadImage**(*file*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.
<i>file</i>	String	Image file name (full path)

4.51 *NewSession Method*

Use the **NewSession** method to initiate a new application session.

Syntax

object.**NewSession**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

4.52 *OpenSession Method*

Use the **OpenSession** method to open a particular session file (full path required).

Syntax

object.**OpenSession**(*session*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>session</i>	String	Session file (full path required)

4.53 *PlayBackward Method*

You can use the **PlayBackward** method to continuously replay the selected disk image(s) backwards.

Syntax

object.**PlayBackward**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.54 *PlayForward Method*

You can use the **PlayForward** method to continuously replay the selected disk image(s) forwards.

Syntax

object.**PlayBackward**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.55 *PlayStop Method*

Use the **PlayStop** method to stop playing disk images.

Syntax

object.**PlayBackward**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.56 *RecordOnImage Method*

Use the **RecordOnImage** method to record a live image.

Syntax

object.**RecordOnImage**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

4

4.57 *Refresh Method*

You can use the **Refresh** method to order a repaint of the IR image.

The method will also make ThermoCAM™ Researcher fully adapt itself to a new image. It involves updating the temperature and frequency tables, scaling conditions and analysis functions.

Call it after you have set all the properties you like and before you start retrieving information from the image.

Syntax

object.**Refresh**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

NOTE: This method will not paint the image itself.

4.58 *RemoveAllAnalysis Method*

You can use the **RemoveAllAnalysis** method to remove all analysis functions from the IR image

Syntax

object.**RemoveAllAnalysis**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Researcher object	For example Session1.

4.59 *RemovePlaybTextCmt Method*

Removes text comments during image playback.

Syntax

Result = *object.RemovePlaybTextCmt(nIndex)*

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>nIndex</i>	The text comment no longer wanted	0,1,2.... -1 will remove all text comments
<i>Result</i>	Boolean	TRUE: Successful outcome.

4.60 *RemovePlaybTextCmtByName Method*

Removes text comments during image playback.

Syntax

Result = *object.RemovePlaybTextCmtByName(strLabel)*

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>strLabel</i>	String	The comment to remove.
<i>Result</i>	Boolean	TRUE: Successful outcome.

4.61 *RemoveRecTextCmt Method*

Removes text comments during image recording.

Syntax

Result = *object.RemoveRecTextCmt(nIndex)*

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>nIndex</i>	The text comment no longer wanted	0,1,2.... -1 will remove all text comments
<i>Result</i>	Boolean	TRUE: Successful outcome.

4.62 *RemoveRecTextCmtByName Method*

Removes text comments during image recording.

Syntax

Result = *object.RemoveRecTextCmtByName(strLabel)*

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>strLabel</i>	String	The comment to remove.
<i>Result</i>	Boolean	TRUE: Successful outcome.

4

4.63 *RestoreAnalysis Method*

You can use the **RestoreAnalysis** method to revert to the original analysis stored in the image.

Syntax

object.RestoreAnalysis

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.64 *RestoreObjParameters Method*

You can use the **RestoreObjParameters** method to revert to the original object parameters stored in the image.

Syntax

object.RestoreObjParameters

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.65 *SaveAreaAsCsv Method*

You can use the **SaveAreaAsCsv** method to save all temperatures in an area or the entire image in a file. A specified separator separates the temperatures and a carriage return linefeed pair separates the lines.

Syntax

temperatures = *object*.**SaveAreaAsCsv**(*filename*, *separator*, *areaFuncName*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>filename</i>	Path and filename for the file where you want the temperatures stored	E.g. "c:\data\area1.csv"
<i>separator</i>	Character that separates the temperatures.	Most often a semicolon is used as a separator, ";".
<i>areaFuncname</i>	Generic name string of an area. An empty string will store the entire image.	AR01
<i>temperatures</i>	Number of values stored in the file. I.e. area width * area height.	

4.66 SaveImage Method

You can use the **SaveImage** method to save the IR image to a file. The format can either be IR Image format (.IMG) or bitmap (.BMP).

Syntax

result = *object*.**SaveImage**(*filename*, *format*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-searcher object	For example Session1.
<i>filename</i>	Full name of the file the image is saved as.	
<i>format</i>	Type of format the image shall be saved as	0 = IR format 1 = BMP format 2 = MatLab 3 = Bitmap with analysis 4 = FLIR public format 5 = CSV 6 = JPEG photo (no IR)
<i>result</i>	Boolean	TRUE if the image was successfully saved FALSE otherwise

NOTE: Changes that you have made to the scaling etc. will be permanently saved in the image when you save it in .IR format and regarded as if they came from the camera when you load it the next time.

4.67 *SaveSession Method*

Saves active session.

Syntax

result = *object*.**SaveSession**(*filename*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>filename</i>	Full name of the file the session is saved as. An empty string saves the session with existing file name	
<i>result</i>	Boolean	TRUE if the session was successfully saved FALSE otherwise

4

4.68 *SetCameraSource Method*

You can use the **SetCameraSource** method to switch between viewing IR images from disk or viewing live images from the camera.

Syntax

object.**SetCameraSource**(*flag*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>flag</i>	Select image source	TRUE = View live images FALSE = View disk images

4.69 *SetFrameRate Method*

Despite its name, this method sets the image rate of the current connected camera.

Syntax

bResult = *object*.**SetFrameRate**(*dbImageRate*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>dbImageRate</i>	New image rate as a double precision float value	100 Hz, for example

Item	Description	Value
<i>bResult</i>	Boolean	TRUE if the rate could be set

NOTE: The image rate will be rounded to the nearest Hz.

NOTE: The method will not succeed if the camera is unable to produce exactly the required image rate.

4.70 *SetInverted Method*

Use the **SetInverted** method to invert the current image palette.

Syntax

object.**SetInverted** (*flag*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>flag</i>	Invert flag	TRUE = Invert current palette FALSE = Normal current palette

4.71 *SetPalette Method*

Use the **SetPalette** method to set a particular colour palette.

Syntax

object.**SetPalette**(*pname*)

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>pname</i>	String, palette name	GREY GREY10 IRON IRON10 MEDICAL RAIN RAIN10 RAIN900

4.72 *SetPlayRate Method*

Set the image playback display rate. A fast forward/backward display rate is achieved by just showing every N:th image. A slow forward/backward display is achieved by an X-fold delay of the image presentation.

Syntax

bResult = *object*.**SetPlayRate**(*bDelay*, *shRate*)

Item	Description	Value
<i>Object</i>	The ThermaCAM™ Re-researcher object	For example Session1.
<i>bDelay</i>	Boolean	TRUE if the image presentation shall be delayed, instead of speeded up.
<i>nRate</i>	Short integer	The N (or X) value
<i>bResult</i>	Boolean	TRUE if successful

4

4.73 SetScaling Method

You can use the **SetScaling** method change the current scale

Syntax

object.**SetScaling**(*scalelimtype*, *min*, *max*, *coldistrtype*)

Item	Description	Value
<i>object</i>	The ThermaCAM™ Re-researcher object	For example Session1.
<i>min,max</i>	The scale minimum and maximum temperatures will be returned here in double precision	15.8 33.6 The temperature is in the current unit.
<i>coldistrtype</i>	The current type of colour distribution is returned here	0 Temperature linear 1 Enhanced contrast (histogram equalization) 2 Camera linear
<i>scalelimtype</i>	The current type of scaling	0 Limits from image 1 Autoadjusted limits 2 Fix limits

NOTE: This method forces ThermaCAM™ Researcher to adapt itself to the current image.

4.74 StepBackward Method

You can use the **StepBackward** method to view the previous disk image.

Syntax

object.**StepBackward**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.75 *StepForward Method*

You can use the **StepForward** method to view the next disk image.

Syntax

4

object.**StepForward**

Item	Description	Value
<i>object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

4.76 *SubtrGetResultImage Method*

May be used after a successfully executed subtraction to return the name of the first subtracted image. If subtraction is put in a sequence file, the name of the sequence file is returned.

Syntax

Result = *object*.**SubtrGetResultImage()**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>Result</i>	String	Name of first subtracted image

SEE ALSO: The **SubtrSaveOutput** Method runs the actual subtraction. For more information, see section:

- 4.77 – SubtrSaveOutput Method on page 70

4.77 *SubtrSaveOutput Method*

Executes subtraction.

Syntax

Result = *object*.**SubtrSaveOutput(RefFileMode)**

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.

Item	Description	Value
<i>RefFileMode</i>	Reference file mode	1 = Consecutive subtraction 2 = First file as reference 3 = Separate reference file
<i>Result</i>	Long	<> 0 if successful

SEE ALSO: For more information, see also the sections:

- 4.81 – SubtrSetReference Method on page 72
- 4.83 – SubtrSetSource Method on page 73
- 4.78 – SubtrSetOptions Method on page 71
- 4.82 – SubtrSetScale Method on page 73
- 4.80 – SubtrSetOutputDir Method on page 72
- 4.79 – SubtrSetOutput Method on page 71
- 4.76 – SubtrGetResultImage Method on page 70

4.78 SubtrSetOptions Method

Sets subtraction offset mode.

Syntax

Result = *object*.**SubtrSetOptions**(*Inverse*, *OpType*, *OffsMode*, *Offset*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>Inverse</i>	BOOL	TRUE = Inverted subtraction FALSE = Normal subtraction
<i>OpType</i>	Long	Not used
<i>OffsMode</i>	Offset mode	<=0 Atmospheric difference as offset =1 Use supplied offset =2 Don't use any offset
<i>Offset</i>	Long	Offset used if <i>OffsMode</i> = 1
<i>Result</i>	Long	<>0 if successful

SEE ALSO: The **SubtrSaveOutput Method** runs the actual subtraction. For more information, see section:

- 4.77 – SubtrSaveOutput Method on page 70

4.79 SubtrSetOutput Method

Syntax

Result = *object*.**SubtrSetOutput**(*Base*, *OneFile*, *OpenSel*, *CloseSrc*, *Palette*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>Base</i>	String	Base name for subtracted files
<i>OneFile</i>	Long	<>0 save in one sequence file else in separate images
<i>OpenSel</i>	Long	<>0 subtraction result will be selected after execution
<i>CloseSrc</i>	Long	<>0 close source files after subtraction
<i>Palette</i>	String	Palette used for subtraction
<i>Result</i>	Long	<>0 if successful

SEE ALSO: The **SubtrSaveOutput Method** runs the actual subtraction. For more information, see section:

- 4.77 – SubtrSaveOutput Method on page 70

4.80 *SubtrSetOutputDir Method*

Sets the output directory for subtracted files.

Syntax

Result = *object*.**SubtrSetOutputDir**(*DirPath*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>DirPath</i>	String	Directory for subtracted images
<i>Result</i>	Long	<>0 if successful

SEE ALSO: The **SubtrSaveOutput Method** runs the actual subtraction. For more information, see section:

- 4.77 – SubtrSaveOutput Method on page 70

4.81 *SubtrSetReference Method*

Set reference image file to be used in subtraction.

Syntax

Result = *object*.**SubtrSetReference**(*FileName*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>FileName</i>	String	Reference image file used for subtraction
<i>Result</i>	Long	<>0 if successful

SEE ALSO: The **SubtrSaveOutput Method** runs the actual subtraction. For more information, see section:

- 4.77 – SubtrSaveOutput Method on page 70

4.82 *SubtrSetScale Method*

Sets Level & Span mode used when subtracting.

Syntax

Result = *object*.**SubtrSetScale**(*DistrModee*, *Low*, *High*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>DistrMode</i>	Level & Span mode	=0 Symmetrical =1 Auto adjust =2 Use supplied <i>High</i> and <i>Low</i> parameters
<i>Low</i>	Long	Lower range limit
<i>High</i>	Long	Upper range limit
<i>Result</i>	Long	<>0 if successful

SEE ALSO: The **SubtrSaveOutput Method** runs the actual subtraction. For more information, see section:

- 4.77 – SubtrSaveOutput Method on page 70

4.83 *SubtrSetSource Method*

Sets source image used for subtraction.

Syntax

Result = *object*.**SubtrSetSource**(*FileName*)

Item	Description	Value
<i>Object</i>	The ThermoCAM™ Re-searcher object	For example Session1.
<i>FileName</i>	String	Source image file

Item	Description	Value
Result	Long	<> 0 if successful

SEE ALSO: The **SubtrSaveOutput Method** runs the actual subtraction. For more information, see section:

- 4.77 – SubtrSaveOutput Method on page 70

Index

A

AddPlaybTextCmt Method: 39
 AddRecTextCmt Method: 39
 AmbientTemperature Property : 5
 AnaDefineFormula Method: 40
 AnaDeleteFormula Method: 40
 AnalysisDefineBentLine Method: 41
 AnalysisDefineBoxArea Method: 42
 AnalysisDefineCircleArea Method: 42
 AnalysisDefineLine Method: 43
 AnalysisDefinePolygonArea Method: 44
 AnalysisDefineSpotmeter Method: 44
 AnalysisDeleteFunc Method: 45
 AnalysisDistance Property: 5
 AnalysisEmissivity Property: 5
 AnalysisFuncNames Method: 45
 AnalysisHistoArray Method: 46
 AnalysisHistoClasses Method: 47
 AnalysisHistoThreshold Method: 47
 AnalysisLabel Property: 6
 AnalysisResults Method: 48
 AnalysisTemperatureArrayLen Method: 49
 AnalysisTemperatureArray Method: 49
 AnalysisTemperatureByName Method: 50
 AnalysisTemperature Property : 6
 AtmosphericTemperature Property : 7
 AutoAdjust Method: 51
 AutoMaxTemperature Property : 8
 AutoMinTemperature Property : 8
 AutoRewind Method: 51

B

BBRRange Property : 9

C

CalcWhileMovingAna Property: 9
 CameraFilter Property : 10
 CameraLens Property : 10
 CameraLevel Property : 10
 Camera Property : 9
 CameraRangeMax Property : 11
 CameraRangeMin Property : 11
 CameraSerialNo Property : 11
 CameraSpan Property : 12
 CameraType Property : 12
 CanHaveBarCodes Property: 12
 Cassette Property : 13
 ComPort Property : 13

Connected Property: 13
 Connect Method: 51
 Cooler Property : 14
 copyright: vi
 CopySessionAndImage Method: 52
 CurrentIROutput Property : 14
 customer support: 1

D

DefaultHistogramClasses Property : 15
 Device Property : 15
 Disconnect Method: 52
 DistanceUnit Property : 15

E

Emissivity Property : 16
 EstimatedTransmission Property : 16
 ExternalCorrection Method: 52
 ExtOpticsTemperature Property : 16
 ExtOpticsTransmission Property : 17

F

FastBackward Method: 52
 FastForward Method: 53
 Filtering Property : 17
 Filter Property : 17
 FLIR Systems
 copyright: vi
 ISO 9001: vi
 legal disclaimer: vi
 product warranty: vi
 quality assurance: vi
 quality management system: vi
 trademarks: vi
 warranty: vi
 FocusFar Method: 53
 FocusNear Method: 53
 FocusStop Method: 53
 Freeze Property : 18

G

GetFormulaByIndex Method: 54
 GetFormulaByName Method: 54
 GetFrameRate Method: 55
 GetNamedValue Method: 55
 GetNoFormulas Method: 55
 GetNumPlaybTextCmts Method: 56
 GetNumRecTextCmts Method: 56

GetPaletteInfo Method : 56
 GetPlaybTextCmtByName Method: 57
 GetPlaybTextCmt Method: 57
 GetPlayRate Method: 58
 GetRecTextCmtByName Method: 58
 GetRecTextCmt Method: 58
 GetScaling Method : 59
 GotoFirstImage Method: 59
 GotoLastImage Method: 60

I

ImageDate Property : 18
 ImageDimensions Method : 60
 ImageFilename Property : 18
 ImageMaxTemperature Property : 19
 ImageMinTemperature Property : 19
 ImagePath Property : 20
 ImageTimeMSec Property : 20
 ImageTime Property : 20
 ImageTimeStr Property : 21
 ImageTitle Property : 21
 InternalCorrection Method: 60
 IsFirstImage Property : 21
 IsLastImage Property : 22
 ISO 9001: vi
 IsUpdated Property: 22

K

KeepAnalysis Property : 22
 KeyTrigF5 Method: 61

L

legal disclaimer: vi
 LoadImage Method: 61

M

methods

AddPlaybTextCmt Method: 39
 AddRecTextCmt Method: 39
 AnaDefineFormula Method: 40
 AnaDeleteFormula Method: 40
 AnalysisDefineBentLine Method: 41
 AnalysisDefineBoxArea Method: 42
 AnalysisDefineCircleArea Method: 42
 AnalysisDefineLine Method: 43
 AnalysisDefinePolygonArea Method: 44
 AnalysisDefineSpotmeter Method: 44
 AnalysisDeleteFunc Method: 45
 AnalysisFuncNames Method : 45
 AnalysisHistoArray Method: 46
 AnalysisHistoClasses Method: 47
 AnalysisHistoThreshold Method: 47

methods (continued)

AnalysisResults Method: 48
 AnalysisTemperatureArrayLen Method: 49
 AnalysisTemperatureArray Method: 49
 AnalysisTemperatureByName Method: 50
 AutoAdjust Method: 51
 AutoRewind Method: 51
 Connect Method: 51
 CopySessionAndImage Method: 52
 Disconnect Method: 52
 ExternalCorrection Method: 52
 FastBackward Method: 52
 FastForward Method: 53
 FocusFar Method: 53
 FocusNear Method: 53
 FocusStop Method: 53
 GetFormulaByIndex Method: 54
 GetFormulaByName Method: 54
 GetFrameRate Method: 55
 GetNamedValue Method: 55
 GetNoFormulas Method: 55
 GetNumPlaybTextCmts Method: 56
 GetNumRecTextCmts Method: 56
 GetPaletteInfo Method : 56
 GetPlaybTextCmtByName Method: 57
 GetPlaybTextCmt Method: 57
 GetPlayRate Method: 58
 GetRecTextCmtByName Method: 58
 GetRecTextCmt Method: 58
 GetScaling Method : 59
 GotoFirstImage Method: 59
 GotoLastImage Method: 60
 ImageDimensions Method : 60
 InternalCorrection Method: 60
 KeyTrigF5 Method: 61
 LoadImage Method: 61
 NewSession Method: 61
 OpenSession Method: 61
 PlayBackward Method: 62
 PlayForward Method: 62
 PlayStop Method: 62
 RecordOnelImage Method: 63
 Refresh Method : 63
 RemoveAllAnalysis Method : 63
 RemovePlaybTextCmtByName Method: 64
 RemovePlaybTextCmt Method: 64
 RemoveRecTextCmtByName Method: 65
 RemoveRecTextCmt Method: 64
 RestoreAnalysis Method : 65
 RestoreObjParameters Method : 65
 SaveAreaAsCsv Method : 65
 SaveImage Method : 66
 SaveSession Method : 67

methods (*continued*)

SetCameraSource Method: 67
 SetFrameRate Method: 67
 SetInverted Method: 68
 SetPalette Method: 68
 SetPlayRate Method: 68
 SetScaling Method: 69
 StepBackward Method: 69
 StepForward Method: 70
 SubtrGetResultImage Method: 70
 SubtrSaveOutput Method: 70
 SubtrSetOptions Method: 71
 SubtrSetOutputDir Method: 72
 SubtrSetOutput Method: 71
 SubtrSetReference Method: 72
 SubtrSetScale Method: 73
 SubtrSetSource Method: 73

Motor Property: 23

N

NewSession Method: 61

O

ObjectDistance Property: 23
 OLELiveLinkDelay Property: 23
 OpenSession Method: 61

P

PlayBackward Method: 62
 PlayForward Method: 62
 PlayStop Method: 62
 PreferredIROutput Property: 24
 product warranty: vi
 properties
 AmbientTemperature Property: 5
 AnalysisDistance Property: 5
 AnalysisEmissivity Property: 5
 AnalysisLabel Property: 6
 AnalysisTemperature Property: 6
 AtmosphericTemperature Property: 7
 AutoMaxTemperature Property: 8
 AutoMinTemperature Property: 8
 BBRange Property: 9
 CalcWhileMovingAna Property: 9
 CameraFilter Property: 10
 CameraLens Property: 10
 CameraLevel Property: 10
 Camera Property: 9
 CameraRangeMax Property: 11
 CameraRangeMin Property: 11
 CameraSerialNo Property: 11
 CameraSpan Property: 12

properties (*continued*)

CameraType Property: 12
 CanHaveBarCodes Property: 12
 Cassette Property: 13
 ComPort Property: 13
 Connected Property: 13
 Cooler Property: 14
 CurrentIROutput Property: 14
 DefaultHistogramClasses Property: 15
 Device Property: 15
 DistanceUnit Property: 15
 Emissivity Property: 16
 EstimatedTransmission Property: 16
 ExtOpticsTemperature Property: 16
 ExtOpticsTransmission Property: 17
 Filtering Property: 17
 Filter Property: 17
 Freeze Property: 18
 ImageDate Property: 18
 ImageFilename Property: 18
 ImageMaxTemperature Property: 19
 ImageMinTemperature Property: 19
 ImagePath Property: 20
 ImageTimeMSec Property: 20
 ImageTime Property: 20
 ImageTimeStr Property: 21
 ImageTitle Property: 21
 IsFirstImage Property: 21
 IsLastImage Property: 22
 IsUpdated Property: 22
 KeepAnalysis Property: 22
 Motor Property: 23
 ObjectDistance Property: 23
 OLELiveLinkDelay Property: 23
 PreferredIROutput Property: 24
 Ratio Property: 24
 RecAutoNameBase Property: 24
 ReclmgPres Property: 25
 RecOneFileRec Property: 25
 RecStartCond Property: 25
 RecStartDelay Property: 26
 RecStartTime Property: 26
 RecStopCond Property: 27
 RecStopDelay Property: 27
 RecStopDuration Property: 27
 RecStopImages Property: 28
 RecStoreCond Property: 28
 RecStoreImages Property: 29
 RecStoreInterval Property: 29
 ReferenceTemperature Property: 29
 RefTemperatureExists Property: 30
 RelativeHumidity Property: 30
 Save900Analysis Property: 30

properties (*continued*)

ShowAnalysisBackColor Property : 31
 ShowAnalysis Property : 31
 ShowAnalysisTemp Property : 32
 ShowIlegTempIndicator Property : 32
 ShowLabels Property: 33
 ShowTemperatureScale Property : 33
 TargetMotion Property : 33
 TemperatureUnit Property : 34
 TotalImages Property : 34
 UseImageObjParams Property : 35
 UseImageScale Property : 35
 UseImageZoom Property : 35
 Version Property: 36
 View3D Property : 36
 Visible Property : 36
 Zoomfactor Property : 37

Q

quality assurance: vi

quality management system: vi

R

Ratio Property : 24
 RecAutoNameBase Property : 24
 RecImgPres Property : 25
 RecOneFileRec Property : 25
 RecordOnImage Method: 63
 RecStartCond Property : 25
 RecStartDelay Property : 26
 RecStartTime Property : 26
 RecStopCond Property : 27
 RecStopDelay Property : 27
 RecStopDuration Property : 27
 RecStopImages Property : 28
 RecStoreCond Property : 28
 RecStoreImages Property : 29
 RecStoreInterval Property : 29
 ReferenceTemperature Property : 29
 Refresh Method : 63
 RefTemperatureExists Property : 30
 RelativeHumidity Property : 30
 RemoveAllAnalysis Method : 63
 RemovePlaybTextCmtByName Method: 64
 RemovePlaybTextCmt Method: 64
 RemoveRecTextCmtByName Method: 65
 RemoveRecTextCmt Method: 64
 RestoreAnalysis Method : 65
 RestoreObjParameters Method : 65

S

Save900Analysis Property : 30

SaveAreaAsCsv Method : 65
 SaveImage Method : 66
 SaveSession Method : 67
 SetCameraSource Method: 67
 SetFrameRate Method: 67
 SetInverted Method: 68
 SetPalette Method: 68
 SetPlayRate Method: 68
 SetScaling Method : 69
 ShowAnalysisBackColor Property : 31
 ShowAnalysis Property : 31
 ShowAnalysisTemp Property : 32
 ShowIlegTempIndicator Property : 32
 ShowLabels Property: 33
 ShowTemperatureScale Property : 33
 StepBackward Method: 69
 StepForward Method: 70
 SubtrGetResultImage Method: 70
 SubtrSaveOutput Method: 70
 SubtrSetOptions Method: 71
 SubtrSetOutputDir Method: 72
 SubtrSetOutput Method: 71
 SubtrSetReference Method: 72
 SubtrSetScale Method: 73
 SubtrSetSource Method: 73
 support, technical: 1

T

TargetMotion Property : 33
 technical support: 1
 TemperatureUnit Property : 34
 TotalImages Property : 34
 trademarks: vi

U

UseImageObjParams Property : 35
 UseImageScale Property : 35
 UseImageZoom Property : 35

V

Version Property: 36
 View3D Property : 36
 Visible Property : 36

W

warranty: vi

Z

Zoomfactor Property : 37

A note on the technical production of this manual

This manual was produced using XML—the *eXtensible Markup Language*. For more information about XML, please visit <http://www.w3.org/XML/>

A note on the typeface used in this manual

This manual was typeset using Swiss 721, which is Bitstream's pan-European version of the Helvetica™ typeface. Helvetica™ was designed by Max Miedinger (1910–1980).

The following file identities and file versions were used in the formatting stream output for this manual:

20236703.xml a40
20242903.xml a2
20243003.xml a3
20243103.xml a4
20243203.xml a6
20254903.xml a55
20287303.xml a6
R0028.rcp a6
config.xml a5



■ AUSTRALIA

FLIR Systems
10 Business Park Drive
Nottinghill
Victoria 3168
Australia
Tel: +61-3-9550-2800
Fax: +61-3-9558-9853
Email: info@flir.com.au
Web: www.flir.com

■ BELGIUM

FLIR Systems
Uitbreidingstraat 60-62
B-2600 Berchem
BELGIUM
Phone: +32 (0)3 287 87 11
Fax: +32 (0)3 287 87 29
E-mail: info@flir.be
Web: www.flir.com

■ BRAZIL

FLIR Systems
Av. Antonio Bardella, 320
CEP: 18085-852 Sorocaba
São Paulo
BRAZIL
Phone: +55 15 3238 8070
Fax: +55 15 3238 8071
E-mail: flir@flir.com.br
Web: www.flir.com

■ CANADA

FLIR Systems
5230 South Service Road, Suite #125
Burlington, ON. L7L 5K2
CANADA
Phone: 1 800 613 0507 ext. 30
Fax: 905 639 5488
E-mail: IRCanada@flir.com
Web: www.flir.com

■ CHINA

FLIR Systems
Beijing Representative Office
Room 509, Building C, Vantone Center
No. A-6 Chaoyangmenwai Ave.
Chaoyang District
Beijing 100020
P.R.C.
Phone: +86 10 5979 7755
Fax: +86 10 8532 2460
E-mail: beijing@flir.com.cn
Web: www.flir.com

■ CHINA

FLIR Systems
Shanghai Representative Office
Room 6311, West Building
Jin Jiang Hotel
59 Maoming Road (South)
Shanghai 200020
P.R.C.
Phone: +86 21 5466 0286
Fax: +86 21 5466 0289
E-mail: shanghai@flir.com.cn
Web: www.flir.com

■ CHINA

FLIR Systems
Guangzhou Representative Office
1105 Main Tower, Guang Dong
International Hotel
339 Huanshi Dong Road
Guangzhou 510098
P.R.C.
Phone: +86 20 8333 7492
Fax: +86 20 8331 0976
E-mail: guangzhou@flir.com.cn
Web: www.flir.com

■ FRANCE

FLIR Systems
10 rue Guynemer
92130 Issy les Moulineaux
Cedex
FRANCE
Phone: +33 (0)1 41 33 97 97
Fax: +33 (0)1 47 36 18 32
E-mail: info@flir.fr
Web: www.flir.com

■ GERMANY

FLIR Systems
Bernier Strasse 81
D-60437 Frankfurt am Main
GERMANY
Phone: +49 (0)69 95 00 900
Fax: +49 (0)69 95 00 9040
E-mail: info@flir.de
Web: www.flir.com

■ GREAT BRITAIN

FLIR Systems
2 Kings Hill Avenue - Kings Hill
West Malling
Kent, ME19 4AQ
UNITED KINGDOM
Phone: +44 (0)1732 220 011
Fax: +44 (0)1732 843 707
E-mail: sales@flir.uk.com
Web: www.flir.com

■ HONG KONG

FLIR Systems
Room 1613-15, Tower 2
Grand Central Plaza
138 Shatin Rural Committee Rd
Shatin, N.T.
HONG KONG
Phone: +852 27 92 89 55
Fax: +852 27 92 89 52
E-mail: flir@flir.com.hk
Web: www.flir.com

■ ITALY

FLIR Systems
Via L. Manara, 2
20051 Limbiate (MI)
ITALY
Phone: +39 02 99 45 10 01
Fax: +39 02 99 69 24 08
E-mail: info@flir.it
Web: www.flir.com

■ JAPAN

FLIR SYSTEMS Japan KK
Nishi-Gotanda Access 8F
3-6-20 Nishi-Gotanda
Shinagawa-Ku
Tokyo 141-0031
JAPAN
Phone: +81 3 6277 5681
Fax: +81 3 6277 5682
E-mail: info@flir.jp
Web: www.flir.com.

■ SWEDEN

FLIR Systems
Worldwide Thermography Center
P.O. Box 3
SE-182 11 Danderyd
SWEDEN
Phone: +46 (0)8 753 25 00
Fax: +46 (0)8 753 23 64
E-mail: sales@flir.se
Web: www.flir.com

■ USA

FLIR Systems
Corporate headquarters
27700A SW Parkway Avenue
Wilsonville, OR 97070
USA
Phone: +1 503 498 3547
Web: www.flir.com

■ USA (Primary sales & service contact in USA)

FLIR Systems
USA Thermography Center
16 Esquire Road
North Billerica, MA. 01862
USA
Phone: +1 978 901 8000
Fax: +1 978 901 8887
E-mail: marketing@flir.com
Web: www.flir.com

■ USA

FLIR Systems
Indigo Operations
70 Castilian Dr.
Goleta, CA 93117-3027
USA
Phone: +1 805 964 9797
Fax: +1 805 685 2711
E-mail: cbi@flir.com
Web: www.corebyindigo.com

■ USA

FLIR Systems
Indigo Operations
IAS Facility
701 John Sims Parkway East
Suite 2B
Niceville, FL 32578
USA
Phone: +1 850 678 4503
Fax: +1 850 678 4992
E-mail: cbi@flir.com
Web: www.corebyindigo.com