

# Integration Objects' Solution for OPC A&E Controls

## OPC A&E ActiveX Version 2.0Rev.2

**User Guide** 

## Compatibility

OPC A&E 1.02 OPC A&E 1.10



Integration Objects' OPC AE ActiveX User's Guide Version 2.0Rev.2 Published May 2018

Copyright © 2004-2018 Integration Objects

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, translated, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Integration Objects. Windows ® and Windows NT ® are registered trademarks of Microsoft Corporation. ActiveX<sup>™</sup> is a trademark of Microsoft Corporation.



## **TABLE OF CONTENTS**

PREF	ACE		14
ABO	UT THIS (	GUIDE	14
AUD	IENCE		14
RELA	TED DOC		14
GETT	ING STA	RTED	16
1.	OVERVI	EW	16
2.	SOFTW	ARE COMPONENTS	16
3.	FEATUR	ES	17
4.	SOFTW	ARE REQUIREMENTS	19
5.	OPC CO	MPATIBILITY	19
6.	INSTALI	ING OPC AE ACTIVEX	19
7.	UNINST	ALLING OPC AF ACTIVEX	25
8	DEPEND		27
			···· 27
1			20
			29
CON	FIGURIIN		วว
	DECION		
1.	DESIGN	-TIME CONFIGURATION	33
1.	DESIGN 1.1.	-TIME CONFIGURATION OPC Options Configuration	<b> 33</b> 33
1.	<b>DESIGN</b> 1.1. 1.2.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration	<b> 33</b> 33 34
1.	<b>DESIGN</b> 1.1. 1.2. 1.3.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration	<b> 33</b> 33 34 35
1.	DESIGN 1.1. 1.2. 1.3. 1.4.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration	<b> 33</b> 33 34 35 36
1.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration	33 33 34 35 36 39
1.	<b>DESIGN</b> 1.1. 1.2. 1.3. 1.4. 1.5. 1.6.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration	33 33 34 35 36 39 41
1.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration Logging	33 33 34 35 36 39 41 43
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration Logging	33 33 34 35 36 39 41 43 45
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration Logging Te CONFIGURATION Connect to an OPC AE Server	33 33 34 35 36 39 41 43 45 46
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration Logging TE CONFIGURATION Connect to an OPC AE Server Disconnect from an AE OPC Server	33 33 34 35 36 39 41 43 43 45 46 47
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3.	-TIME CONFIGURATION	33 33 34 35 36 39 41 43 43 45 46 47 47
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4.	-TIME CONFIGURATION	33 33 34 35 36 39 41 43 43 45 46 47 47 49
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration Logging ME CONFIGURATION Connect to an OPC AE Server Disconnect from an AE OPC Server Acknowledge Alarms Server Information Event Subscription Status	33 33 34 35 36 39 41 43 45 45 47 47 49 49
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5. 2.6.	-TIME CONFIGURATION OPC Options Configuration OPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration Logging TE CONFIGURATION Connect to an OPC AE Server Disconnect from an AE OPC Server Acknowledge Alarms Server Information Event Subscription Status Select Returned Attributes	33 33 34 35 36 39 41 43 43 45 47 47 49 49 50
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7	-TIME CONFIGURATION OPC Options Configuration	33 33 34 35 36 39 41 43 43 45 47 47 49 49 50 51
1. 2.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7. 2.8	-TIME CONFIGURATION	33 33 34 35 36 39 41 43 43 45 47 47 49 50 51
1.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7. 2.8. 2.9	-TIME CONFIGURATION         OPC Options Configuration         OPC Alarm Color Configuration         Email Alarms Configuration         Control Layout Configuration         Filter Constraints Configuration         Save Alarm Reports Configuration         Logging         ME CONFIGURATION         Connect to an OPC AE Server.         Disconnect from an AE OPC Server.         Acknowledge Alarms         Server Information         Event Subscription Status         Select Returned Attributes         Retrieve Returned Attributes         Enable/Disable condition by areas and sources	33 33 34 35 36 39 41 43 43 45 46 47 47 49 50 51 52
1.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7. 2.8. 2.9. 2.10	-TIME CONFIGURATION OPC Options Configuration DPC Alarm Color Configuration Email Alarms Configuration Control Layout Configuration Filter Constraints Configuration Save Alarm Reports Configuration Logging Connect to an OPC AE Server. Disconnect from an AE OPC Server Acknowledge Alarms Server Information Event Subscription Status Select Returned Attributes Retrieve Returned Attributes Enable/Disable condition by areas and sources Available Filters OPC Eithering Configuration	33 34 35 36 39 41 43 43 45 47 47 49 50 51 52 54
1.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7. 2.8. 2.9. 2.10. 2.11	-TIME CONFIGURATION	33 33 34 35 36 39 41 43 43 43 45 47 49 49 50 51 52 54 55
1.	DESIGN 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 1.7. RUNTIN 2.1. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7. 2.8. 2.9. 2.10. 2.11. 2.21. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7. 2.8. 2.9. 2.10. 2.11. 2.2. 2.3. 2.4. 2.5. 2.6. 2.7. 2.8. 2.9. 2.10	-TIME CONFIGURATION         OPC Options Configuration         OPC Alarm Color Configuration         Email Alarms Configuration         Email Alarms Configuration         Control Layout Configuration         Filter Constraints Configuration         Save Alarm Reports Configuration         Logging         Me CONFIGURATION         Connect to an OPC AE Server.         Disconnect from an AE OPC Server.         Acknowledge Alarms         Server Information         Event Subscription Status         Select Returned Attributes         Retrieve Returned Attributes         Enable/Disable condition by areas and sources         Available Filters         OPC Filtering Configuration	33 33 34 35 36 39 41 43 43 45 47 47 49 50 51 52 54 55



4.         CONTROL EVENTS         60           4.1.         DREvent ovent Vent (OpeEvent as Object)         60           4.2.         Public Event On Event (OpeEvent as Object)         60           4.2.         Public Porentry Activities As Boolian         60           4.2.         Public Porentry Activities As Boolian         60           4.2.         Public Porentry Activities As Date         60           4.2.3.         Public Porentry ConstronName As Strinks         60           4.2.4.         Public Porentry ConstronName As Strinks         60           4.2.5.         Public Porentry ConstronName As Strinks         60           4.2.6.         Public Porentry VisinCateown A Long         60           4.2.7.         Public Porentry Strinks Long         61           4.2.8.         Public Porentry Strinks Long         61           4.2.10.         Public Porentry Strinks Long         62           4.2.11.         Public Porentry Strinks Long         62           4.2.12.         Public Porentry Conster As Strinks         62           4.2.14.         Public Porentry Strinks Long         62           4.2.15.         Public Porentry Eventry As Long         62           4.2.16.         Public Porentry Eventry Thetrinks Long         62 </th <th>3.</th> <th>CONT</th> <th>IROL PROPERTIES AND METHODS</th> <th>59</th>	3.	CONT	IROL PROPERTIES AND METHODS	59
4.1       OnEvent event       60         4.2       Public Propert VeckReguineo As Boolean       60         4.2.1.       Public Propert VeckReguineo As Boolean       60         4.2.2.       Public Propert VeckReguineo As Date       60         4.2.3.       Public Propert VeckReguineo As Date       60         4.2.4.       Public Propert VeckReguineo As Straing       60         4.2.5.       Public Propert VeckReguineo As Straing       60         4.2.6.       Public Propert VeckReguineo As Straing       60         4.2.7.       Public Propert VeckReguineo As Straing       60         4.2.8.       Public Propert VeckReguineo As Straing       60         4.2.9.       Public Propert VeckReguineo As Straing       61         4.2.10.       Public Propert Vick VeckReguineo As Innig       61         4.2.11.       Public Propert Vick VeckReguineo As Innig       61         4.2.12.       Public Propert Vick VeckReguineo As Innig       62         4.2.13.       Public Propert VeckReguineo As Innig       62         4.2.14.       Public Propert VeckReguineo As Long       62         4.2.15.       Public Propert VeckReguineo As Innig       62         4.2.16.       Public Propert VeckReguineo As Long       62         4.2.17.	4.	CON	IROL EVENTS	60
4.2.       Public Event OnEvent(OptEvent as Object)       60         4.2.1.       Pusut PROPERTY ACREMOUND AS BODELAN       60         4.2.2.       Pusut PROPERTY ACREMOUND AS BODELAN       60         4.2.4.       Pusut PROPERTY EVENTTIME AS DATE       60         4.2.4.       Pusut PROPERTY SUECONDITIONNAME AS STRING       60         4.2.5.       Pusut PROPERTY EVENTATESORY AS LONG       60         4.2.6.       Pusut PROPERTY EVENTATESORY AS LONG       60         4.2.7.       Pusut PROPERTY CONK AS LONG       60         4.2.8.       Pusut PROPERTY TOONE AS LONG       60         4.2.9.       Pusut PROPERTY STATES LONG       61         4.2.10.       Pusut PROPERTY STATES LONG       61         4.2.11.       Pusut PROPERTY STATES LONG       62         4.2.12.       Pusut PROPERTY STATES LONG       62         4.2.13.       Pusut PROPERTY STATES LONG       62         4.2.14.       Pusut PROPERTY STOREON STRING       62         4.2.15.       Pusut PROPERTY STOREON STRING       62         4.2.14.       Pusut PROPERTY STOREONES AS LONG       62         4.2.15.       Pusut PROPERTY EVENTATES LONG       62         4.2.16.       PUSUT PROPERTY CHART AS LONG       62 <t< td=""><td></td><td>4.1.</td><td>OnEvent event</td><td> 60</td></t<>		4.1.	OnEvent event	60
4.2.1       Public PROPERTY ACKREQUIED AS BOOLEAN       60         4.2.2       Public PROPERTY CUPIENTE AS DATE.       60         4.2.3       Public PROPERTY CONTITION NAME AS STRING       60         4.2.4       Public PROPERTY SUPERTY CONTITION NAME AS STRING       60         4.2.5       Public PROPERTY SUPERCATEGORY AS LONG.       60         4.2.6       Public PROPERTY VENTATE AS LONG.       60         4.2.7       Public PROPERTY MESSAGE AS STRING.       60         4.2.8       Public PROPERTY MESSAGE AS STRING.       60         4.2.9       Public PROPERTY VENTATE AS LONG.       61         4.2.10       Public PROPERTY STRQUALITY AS STRING.       61         4.2.11       Public PROPERTY STRQUALITY AS STRING.       62         4.2.12       Public PROPERTY VENT TRA SLONG.       62         4.2.13       Public PROPERTY VENT TRA SLONG.       62         4.2.14       Public PROPERTY VENT TRA SLONG.       62         4.2.15       Public PROPERTY CONIDIA STRING       62         4.2.16       Public PROPERTY CONID AS STRING       62         4.2.15       Public PROPERTY CONID AS STRING       62         4.2.16       Public PROPERTY CONID AS STRING       62         4.2.17       Public PROPERTY CONERD AS STRING </td <td></td> <td>4.2.</td> <td>Public Event OnEvent(OpcEvent as Object)</td> <td> 60</td>		4.2.	Public Event OnEvent(OpcEvent as Object)	60
4.2.2.       Public PROPERTY VENTIMIE AS DATE.       60         4.2.3.       Public PROPERTY VENTIME AS DATE.       60         4.2.4.       Public PROPERTY VENTIME AS DATE.       60         4.2.5.       Public PROPERTY VENTIME AS DATE.       60         4.2.6.       Public PROPERTY CONDINAME AS STRING.       60         4.2.6.       Public PROPERTY CONCERAS LONG.       60         4.2.7.       Public PROPERTY MEXSTARE AS LONG.       60         4.2.9.       Public PROPERTY MESSTARE AS LONG.       61         4.2.10.       Public PROPERTY NUESTARE AS LONG.       61         4.2.11.       Public PROPERTY SUCALITY AS STRING.       61         4.2.12.       Public PROPERTY SUVERTY AS LONG.       62         4.2.13.       Public PROPERTY SUVERTY AS LONG.       62         4.2.14.       Public PROPERTY SUVERTY AS LONG.       62         4.2.15.       Public PROPERTY CONCIMAS STRING       62         4.2.16.       Public PROPERTY VENTATTRIBUTES LONG.       62         4.2.17.       Public PROPERTY VENTATTRIBUTES LONG.       62         4.2.18.       Public PROPERTY EVENTATTRIBUTES LONG.       63         4.2.19.       Public PROPERTY EVENTATTRIBUTES LONG.       63         4.2.117.       Public PROPERTY EVENTATTRIBUT	4.2.1	. Pu	IBLIC PROPERTY ACKREQUIRED AS BOOLEAN	60
4.2.3.       PUBLIC PROPERT Y CONTONNAME AS STRING       60         4.2.4.       PUBLIC PROPERTY SUBCONDITIONNAME AS STRING       60         4.2.5.       PUBLIC PROPERTY SUBCONDITIONNAME AS STRING       60         4.2.7.       PUBLIC PROPERTY SUBCONDITIONNAME AS STRING       60         4.2.7.       PUBLIC PROPERTY CONTRACTORY AS LONG       60         4.2.8.       PUBLIC PROPERTY SUBCONDITIONNAME AS STRING       60         4.2.9.       PUBLIC PROPERTY NEWSTATE AS LONG       61         4.2.10.       PUBLIC PROPERTY STRUQUALTY AS STRING       61         4.2.11.       PUBLIC PROPERTY SUBCONDERTY SUBCONDERTY STRUGUALTY AS STRING       62         4.2.12.       PUBLIC PROPERTY SUBCONDERTY SUBCOND	4.2.2	. Pu	IBLIC PROPERTY ACTIVETIME AS DATE	60
4.2.4.       PUBLIC PROPERTY SUPERTY CONDITION NAME AS STRING       60         4.2.5.       PUBLIC PROPERTY EVENTCATEGORY AS LONG.       60         4.2.6.       PUBLIC PROPERTY EVENTCATEGORY AS LONG.       60         4.2.7.       PUBLIC PROPERTY EVENTCATEGORY AS LONG.       60         4.2.8.       PUBLIC PROPERTY NEWSTATE AS LONG.       60         4.2.9.       PUBLIC PROPERTY QUALITY AS LONG.       61         4.2.10.       PUBLIC PROPERTY SEVENTY AS LONG.       61         4.2.11.       PUBLIC PROPERTY SEVENTY AS LONG.       62         4.2.12.       PUBLIC PROPERTY SEVENTY AS LONG.       62         4.2.13.       PUBLIC PROPERTY SEVENTY SEVENTS       62         4.2.14.       PUBLIC PROPERTY VENTY TYPE AS LONG.       62         4.2.15.       PUBLIC PROPERTY VENTY TATSIBUTES COUNT AS LONG.       62         4.2.16.       PUBLIC PROPERTY VENTATTRIBUTES (UNDEX AS INTEGER) AS VARIANT.       63         CONFIGURING THE INET AELOGGER.       64       64         1.       RUNTIME CONFIGURATION       64         4.3.       View.       65         4.4.       Auto Load Configuration.       65         4.5.       Load Configuration.       66         4.1.       Clean Configuration.       67	4.2.3	. Pu	IBLIC PROPERTY EVENTTIME AS DATE	60
4.2.5.       PUBLIC PROPERTY SUNCANTIONNAME AS TRINKIG       60         4.2.6.       PUBLIC PROPERTY COXKE AS LONG       60         4.2.7.       PUBLIC PROPERTY COXKE AS LONG       60         4.2.7.       PUBLIC PROPERTY COXKE AS LONG       61         4.2.9.       PUBLIC PROPERTY QUALITY AS LONG       61         4.2.10.       PUBLIC PROPERTY VENTATE AS LONG       61         4.2.11.       PUBLIC PROPERTY SEVENTA S LONG       62         4.2.12.       PUBLIC PROPERTY SEVENTA S LONG       62         4.2.13.       PUBLIC PROPERTY SEVENTA SCHONG       62         4.2.14.       PUBLIC PROPERTY EVENTATE AS LONG       62         4.2.15.       PUBLIC PROPERTY CHANGEMASK AS LONG       62         4.2.16.       PUBLIC PROPERTY EVENTATTIBUITES COUNT AS LONG       63         4.2.17.       PUBLIC PROPERTY EVENTATTIBUITES COUNT AS LONG       63         CONFIGURING THE .NET AE LOGGER       64       64         1.       RUNTIME CONFIGURATION       65         4.3.       View       65         4.4.       Auto Load Configuration       66         4.1.       Clean Colfguration       66         4.2.       Connect to an AE OPC Server       67         4.3.       View <td< td=""><td>4.2.4</td><td>. Pu</td><td>IBLIC PROPERTY CONDITIONNAME AS STRING</td><td>60</td></td<>	4.2.4	. Pu	IBLIC PROPERTY CONDITIONNAME AS STRING	60
4.2.6.       PUBLIC PROPERTY EVENTCATEGORY AS LONG.       60         4.2.7.       PUBLIC PROPERTY MESSAGE AS STRING.       60         4.2.8.       PUBLIC PROPERTY MESSAGE AS STRING.       60         4.2.10.       PUBLIC PROPERTY MESSAGE AS STRING.       61         4.2.10.       PUBLIC PROPERTY MESSAGE AS STRING.       61         4.2.11.       PUBLIC PROPERTY STQUALITY AS STRING.       61         4.2.12.       PUBLIC PROPERTY SEVENTY AS LONG.       62         4.2.13.       PUBLIC PROPERTY SEVENTY SOURCE AS STRING.       62         4.2.14.       PUBLIC PROPERTY EVENTY TYPE AS LONG.       62         4.2.15.       PUBLIC PROPERTY CHANCEMASK AS LONG.       62         4.2.16.       PUBLIC PROPERTY EVENTATTRIBUTESCOUNT AS LONG.       63         4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTESCOUNT AS LONG.       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES(INDEX AS INTEGER) AS VARIANT.       63         CONFIGURING THE NET AE LOGGER       64       64         1.       RUNTIME CONFIGURATION       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration	4.2.5	. Pu	IBLIC PROPERTY SUBCONDITIONNAME AS STRING	60
4.2.7.       PUBLIC PROPERTY COVIE AS LONG.       60         4.2.8.       PUBLIC PROPERTY MESSAGE AS STRING.       60         4.2.9.       PUBLIC PROPERTY NEWSTATE AS LONG.       61         4.2.10.       PUBLIC PROPERTY NEWSTATE AS LONG.       61         4.2.11.       PUBLIC PROPERTY SEQUALITY AS STRING.       61         4.2.11.       PUBLIC PROPERTY SEQUALITY AS LONG.       62         4.2.12.       PUBLIC PROPERTY SEQUEL AS STRING.       62         4.2.13.       PUBLIC PROPERTY CHANCE AS STRING.       62         4.2.14.       PUBLIC PROPERTY CHANCE AS STRING.       62         4.2.15.       PUBLIC PROPERTY CHANCE AS STRING.       62         4.2.16.       PUBLIC PROPERTY CHANCE MAS STRING.       62         4.2.17.       PUBLIC PROPERTY CHANCEMASK AS LONG.       62         4.2.18.       PUBLIC PROPERTY CHANCEMASK AS LONG.       63         CONFIGURING THE .NET AE LOGGER       64       64         1.       RUNTIME CONFIGURATION       64         4.3.       View.       65       4.4.         4.4.       Auto Load Configuration.       65         4.5.       Load Configuration.       66         4.1.       Clean Configuration.       66         4.2.       C	4.2.6	. Pu	IBLIC PROPERTY EVENTCATEGORY AS LONG	60
4.2.8.       PUBLIC PROPERTY MESSAGE AS STRING       60         4.2.9.       PUBLIC PROPERTY NEWSTATE AS LONG.       61         4.2.10.       PUBLIC PROPERTY STRUALITY AS LONG.       61         4.2.11.       PUBLIC PROPERTY STRUALITY AS LONG.       62         4.2.12.       PUBLIC PROPERTY SUVERITY AS LONG.       62         4.2.13.       PUBLIC PROPERTY SOURCE AS STRING       62         4.2.14.       PUBLIC PROPERTY ACTORID AS STRING       62         4.2.15.       PUBLIC PROPERTY ACTORID AS STRING       62         4.2.16.       PUBLIC PROPERTY CHANGEMASK AS LONG.       62         4.2.17.       PUBLIC PROPERTY CHANGEMASK AS LONG.       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTESCOUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATRIBUTESCOUNT AS LONG       63         CONFIGURING THE .NET AE LOGGER.       64       63         1.       RUNTIME CONFIGURATION       64         4.3.       View       65         4.4.       Auto Load Configuration       66         4.5.       Load Configuration       66         4.6.       Connect troan AE OPC Server.       67         4.7.       Connect troan AE OPC Server.       67         4.8.       Ret	4.2.7	. Pu	IBLIC PROPERTY COOKIE AS LONG	60
4.2.9.       PUBLIC PROPERTY DUALITY AS LONG.       61         4.2.10.       PUBLIC PROPERTY GUALITY AS STRING.       61         4.2.11.       PUBLIC PROPERTY STRQUALITY AS STRING.       62         4.2.12.       PUBLIC PROPERTY SURCE AS STRING.       62         4.2.13.       PUBLIC PROPERTY SURTY EAS LONG.       62         4.2.14.       PUBLIC PROPERTY EVENT FYE AS LONG.       62         4.2.15.       PUBLIC PROPERTY CHANGE MAS KAS LONG.       62         4.2.16.       PUBLIC PROPERTY EVENTATRIBUTESCOURT AS LONG.       63         4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTESCOURT AS LONG.       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES (INDEX AS INTEGER) AS VARIANT.       63         CONFIGURING THE .NET AE LOGGER.       64       64         1.       RUNTIME CONFIGURATION       64         4.3.       View.       65         4.4.       Auto Load Configuration       66         4.2.       Connect to an AE OPC Server.       67         4.3.       Disconnect from an AE OPC Server.       67         4.4.       Acknowledging alarms.       67         4.5.       Server Information       69         4.6.       Event Subscription Status.       70         4.7.	4.2.8	. Pu	IBLIC PROPERTY MESSAGE AS STRING	60
4.2.10.       PUBLIC PROPERTY SUQUALITY AS LONG.       61         4.2.11.       PUBLIC PROPERTY STRQUALITY AS STRING       61         4.2.12.       PUBLIC PROPERTY SEVERITY AS LONG.       62         4.2.13.       PUBLIC PROPERTY EVENT TYPE AS LONG.       62         4.2.14.       PUBLIC PROPERTY EVENTTYPE AS LONG.       62         4.2.15.       PUBLIC PROPERTY ACTORID AS STRING       62         4.2.16.       PUBLIC PROPERTY ACTORID AS STRING       62         4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTES(LOUT AS LONG       63         CONFIGURATION THE VENTATTRIBUTES(INDEX AS INTEGER) AS VARIANT.       63         CONFIGURATION THE AL LOGGER       64         1.       RUNTIME CONFIGURATION       65         4.3.       View.       65         4.4.       Auto Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect from an AE OPC Server.       67         4.4.       Acknowledging alarms       67         4.5.       Load Configuration       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returmed Attributes       72 <t< td=""><td>4.2.9</td><td>. Pu</td><td>IBLIC PROPERTY NEWSTATE AS LONG</td><td>61</td></t<>	4.2.9	. Pu	IBLIC PROPERTY NEWSTATE AS LONG	61
4.2.11.       PUBLIC PROPERTY STRUALITY AS STRING       61         4.2.12.       PUBLIC PROPERTY SEVENTY AS LONG       62         4.2.13.       PUBLIC PROPERTY SURCE AS STRING       62         4.2.14.       PUBLIC PROPERTY SURCE AS STRING       62         4.2.15.       PUBLIC PROPERTY ACTORID AS STRING       62         4.2.16.       PUBLIC PROPERTY CHANGE MASK AS LONG       62         4.2.17.       PUBLIC PROPERTY CHANGE MASK AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES COUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES (INDEX AS INTEGER) AS VARIANT       63         CONFIGURING THE .NET AE LOGGER       64       64         1.       RUNTIME CONFIGURATION       64         4.3.       View       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       67         4.2.       Connect to an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes	4.2.1	0.	Public Property Quality As Long	61
4.2.12.       PUBLIC PROPERTY SEVENTY AS LONG       62         4.2.13.       PUBLIC PROPERTY SOURCE AS STRING       62         4.2.14.       PUBLIC PROPERTY EVENTTYPE AS LONG       62         4.2.15.       PUBLIC PROPERTY EVENTTYPE AS LONG       62         4.2.17.       PUBLIC PROPERTY EVENTATRIBUTESCOUNT AS LONG       63         4.2.17.       PUBLIC PROPERTY EVENTATRIBUTESCOUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATRIBUTES(INDEX AS INTEGER) AS VARIANT       63         CONFIGURING THE .NET AE LOGGER       64       64         1.       RUNTIME CONFIGURATION       64         4.3.       View       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       72         4.8.       Retrieve Returned Attributes       72	4.2.1	1.	Public Property StrQuality As String	61
4.2.13.       PUBLIC PROPERTY SURCE AS STRING       62         4.2.14.       PUBLIC PROPERTY EVENTTYPE AS LONG       62         4.2.15.       PUBLIC PROPERTY ACTORID AS STRING       62         4.2.16.       PUBLIC PROPERTY CHANGEMASK AS LONG       62         4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTES(COUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES(INDEX AS INTEGER) AS VARIANT       63         CONFIGURING THE .NET AE LOGGER       64         1.       RUNTIME CONFIGURATION       64         4.3.       View       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.4.       Acknowledging alarms       70         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72 <t< td=""><td>4.2.1</td><td>2.</td><td>Public Property Severity As Long</td><td>62</td></t<>	4.2.1	2.	Public Property Severity As Long	62
4.2.14.       PUBLIC PROPERTY EVENTTYPE AS LONG       62         4.2.15.       PUBLIC PROPERTY CATORID AS STRING       62         4.2.16.       PUBLIC PROPERTY CHANCEMASK AS LONG       62         4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTESCOUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES(INDEX AS INTEGER) AS VARIANT       63         CONFIGURING THE .NET AE LOGGER       64         1.       RUNTIME CONFIGURATION       64         4.3.       View.       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server.       67         4.3.       Disconnect from an AE OPC Server.       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters       78	4.2.1	3.	Public Property Source As String	62
4.2.15.       PUBLIC PROPERTY CORID AS STRING       62         4.2.16.       PUBLIC PROPERTY CHANGEMASK AS LONG       63         4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTES COUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES (INDEX AS INTEGER) AS VARIANT.       63         CONFIGURING THE .NET AE LOGGER       64         1.       RUNTIME CONFIGURATION       64         4.3.       View       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server.       67         4.3.       Disconnect from an AE OPC Server.       67         4.4.       Acknowledging alarms       67         4.5.       Server information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters       72         4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       78         4.13.	4.2.1	4.	PUBLIC PROPERTY EVENTTYPE AS LONG	62
4.2.16.       PUBLIC PROPERTY CHANGEMASK AS LONG.       62         4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTESCOUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES(INDEX AS INTEGER) AS VARIANT.       63         CONFIGURING THE .NET AE LOGGER       64         1.       RUNTIME CONFIGURATION       64         4.3.       View.       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server.       67         4.3.       Disconnect from an AE OPC Server.       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources.       72         4.10.       Available filters.       78         4.11.       OPC Filtering Configuration       79         4.12.       Apply Filter Constraints       78         4.13.	4.2.1	5.	Public Property ActorID As String	62
4.2.17.       PUBLIC PROPERTY EVENTATTRIBUTESCOUNT AS LONG       63         4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES(INDEX AS INTEGER) AS VARIANT.       63         CONFIGURING THE .NET AE LOGGER       64         1.       RUNTIME CONFIGURATION       64         4.3.       View	4.2.1	6.	Public Property ChangeMask As Long	62
4.2.18.       PUBLIC PROPERTY EVENTATTRIBUTES (INDEX AS INTEGER) AS VARIANT.       63         CONFIGURING THE .NET AE LOGGER       64         1.       RUNTIME CONFIGURATION       64         4.3.       View.       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters       74         4.11.       OPC Filtering Constraints       78         4.13.       Clear View.       79         4.14.       Print.       79         4.15.       Save       82         4.10.       Available filters       82         4.11.	4.2.1	7.	PUBLIC PROPERTY EVENTATTRIBUTESCOUNT AS LONG	63
CONFIGURING THE .NET AE LOGGER641.RUNTIME CONFIGURATION644.3.View654.4.Auto Load Configuration654.5.Load Configuration664.1.Clean Configuration664.2.Connect to an AE OPC Server674.3.Disconnect from an AE OPC Server674.4.Acknowledging alarms674.5.Server Information694.6.Event Subscription Status704.7.Select Returned Attributes714.8.Retrieve Returned Attributes724.9.Enable/Disable condition by areas and sources724.10.Available filters744.11.OPC Filtering Configuration754.12.Apply Filter Constraints784.13.Clear View794.14.Print794.15.Sove824.16.POP OPTIES824.16.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION834.16.4.CONTROL LAYOUT CONFIGURATION834.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.Save ALARM REPORTS CONFIGURATION884.16.5.FILTER CONSTRAINTS CONFIGURATION88	4.2.1	8.	PUBLIC PROPERTY EVENTATTRIBUTES (INDEX AS INTEGER) AS VARIANT	63
1.       RUNTIME CONFIGURATION       64         4.3.       View       65         4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters.       74         4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       78         4.13.       Clear View       79         4.14.       Print       79         4.15.       Save       82         4.16.       Properties       83         4.16.1.       OPC Configuration<	CON	FIGUR	ING THE .NET AE LOGGER	64
4.3.       View	1.	RUN	FIME CONFIGURATION	64
4.4.       Auto Load Configuration       65         4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server.       67         4.3.       Disconnect from an AE OPC Server.       67         4.3.       Disconnect from an AE OPC Server.       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters       74         4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       78         4.13.       Clear View       79         4.14.       Print       79         4.15.       Save       82         4.16.1.       OPC OPTIONS CONFIGURATION       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.3.       EMAIL ALARMS CONFIGURATION       83		4.3.	View	
4.5.       Load Configuration       66         4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters       74         4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       78         4.13.       Clear View       79         4.14.       Print       79         4.15.       Save       82         4.16.1.       OPC OPTIONS CONFIGURATION       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.3.       EMAIL ALARMS CONFIGURATION       83         4.16.4.       CONTROL LAYOUT CONFIGURATION       85         4.16.5.       FILTER CONSTRAINTS CONFIGURATION       85     <		4.4.	Auto Load Configuration	
4.1.       Clean Configuration       66         4.2.       Connect to an AE OPC Server       67         4.3.       Disconnect from an AE OPC Server       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters.       74         4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       79         4.13.       Clear View.       79         4.14.       Print.       79         4.15.       Save       82         4.16.       OPC OPTIONS CONFIGURATION       83         4.16.1.       OPC CONFIGURATION       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.3.       EMAIL ALARMS CONFIGURATION       84         4.16.4.       CONTROL LAYOUT CONFIGURATION       84         4.16.5.       FILTER CONSTRAINTS CONFIGURATION       88		4.5.	Load Configuration	
4.2.       Connect to an AE OPC Server.       67         4.3.       Disconnect from an AE OPC Server.       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters       74         4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       78         4.13.       Clear View       79         4.14.       Print       79         4.15.       Save       82         4.16.       OPC OPTIONS CONFIGURATION       83         4.16.1.       OPC ALARM COLOR CONFIGURATION       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.3.       EMAIL ALARMS CONFIGURATION       84         4.16.4.       CONTROL LAYOUT CONFIGURATION       85         4.16.5.       FILTER CONSTRAINTS CONFIGURATION       88         4.16.6.       SAVE ALARM REPORTS CONFIGURATIO		4.1.	Clean Configuration	
4.3.       Disconnect from an AE OPC Server.       67         4.4.       Acknowledging alarms       67         4.5.       Server Information       69         4.6.       Event Subscription Status       70         4.7.       Select Returned Attributes       71         4.8.       Retrieve Returned Attributes       72         4.9.       Enable/Disable condition by areas and sources       72         4.10.       Available filters       74         4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       78         4.13.       Clear View       79         4.14.       Properties       82         4.16.       OPC OPTIONS CONFIGURATION       83         4.16.1.       OPC ALARM COLOR CONFIGURATION       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.4.       CONTROL LAYOUT CONFIGURATION       84         4.16.5.       FILTER CONSTRAINTS CONFIGURATION       85         4.16.6.       SAVE ALARM REPORTS CONFIGURATION       88		4.2.	Connect to an AE OPC Server	
4.4.Acknowledging alarms674.5.Server Information694.6.Event Subscription Status704.7.Select Returned Attributes714.8.Retrieve Returned Attributes724.9.Enable/Disable condition by areas and sources724.10.Available filters744.11.OPC Filtering Configuration754.12.Apply Filter Constraints784.13.Clear View794.14.Print794.15.Save824.16.Properties824.16.OPC OPTIONS CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION834.16.4.CONTROL LAYOUT CONFIGURATION844.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION88		4.3.	Disconnect from an AE OPC Server	
4.5.Server Information694.6.Event Subscription Status704.7.Select Returned Attributes714.8.Retrieve Returned Attributes724.9.Enable/Disable condition by areas and sources724.10.Available filters744.11.OPC Filtering Configuration754.12.Apply Filter Constraints784.13.Clear View794.14.Print794.15.Save824.16.Properties824.16.1.OPC Configuration834.16.2.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION844.16.4.CONTROL LAYOUT CONFIGURATION854.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.Save ALARM REPORTS CONFIGURATION88		4.4.	Acknowledging alarms	
4.6.Event Subscription Status704.7.Select Returned Attributes714.8.Retrieve Returned Attributes724.9.Enable/Disable condition by areas and sources724.10.Available filters744.11.OPC Filtering Configuration754.12.Apply Filter Constraints784.13.Clear View794.14.Print794.15.Save824.16.1.OPC OPTIONS CONFIGURATION834.16.2.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION844.16.4.CONTROL LAYOUT CONFIGURATION884.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION80		4.5.	Server Information	
4.7.Select Returned Attributes714.8.Retrieve Returned Attributes724.9.Enable/Disable condition by areas and sources724.10.Available filters744.11.OPC Filtering Configuration754.12.Apply Filter Constraints784.13.Clear View794.14.Print794.15.Save824.16.Properties824.16.1.OPC OPTIONS CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION834.16.4.CONTROL LAYOUT CONFIGURATION844.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION88		4.6.	Event Subscription Status	
4.8.Retrieve Returned Attributes.724.9.Enable/Disable condition by areas and sources724.10.Available filters.744.11.OPC Filtering Configuration.754.12.Apply Filter Constraints784.13.Clear View.794.14.Print794.15.Save824.16.Properties824.16.1.OPC OPTIONS CONFIGURATION834.16.2.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION844.16.4.CONTROL LAYOUT CONFIGURATION854.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION80		4.7.	Select Returned Attributes	
4.9.Enable/Disable condition by areas and sources724.10.Available filters744.11.OPC Filtering Configuration754.12.Apply Filter Constraints784.13.Clear View794.14.Print794.15.Save824.16.Properties824.16.1.OPC OPTIONS CONFIGURATION834.16.2.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION844.16.4.CONTROL LAYOUT CONFIGURATION854.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION90		4.8.	Retrieve Returned Attributes	
4.10.Available filters.744.11.OPC Filtering Configuration754.12.Apply Filter Constraints784.13.Clear View.794.14.Print794.15.Save824.16.Properties.824.16.1.OPC OPTIONS CONFIGURATION834.16.2.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION844.16.4.CONTROL LAYOUT CONFIGURATION854.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION90		4.9.	Enable/Disable condition by areas and sources	
4.11.       OPC Filtering Configuration       75         4.12.       Apply Filter Constraints       78         4.13.       Clear View       79         4.14.       Print       79         4.15.       Save       82         4.16.       Properties       82         4.16.1.       OPC OPTIONS CONFIGURATION       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.3.       EMAIL ALARMS CONFIGURATION       83         4.16.4.       CONTROL LAYOUT CONFIGURATION       85         4.16.5.       FILTER CONSTRAINTS CONFIGURATION       88         4.16.6.       SAVE ALARM REPORTS CONFIGURATION       890		4.10.	Available filters	
4.12.       Apply Filter Constraints       78         4.13.       Clear View       79         4.14.       Print       79         4.15.       Save       82         4.16.       Properties       82         4.16.1.       OPC OPTIONS CONFIGURATION       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.3.       EMAIL ALARMS CONFIGURATION       83         4.16.4.       CONTROL LAYOUT CONFIGURATION       85         4.16.5.       FILTER CONSTRAINTS CONFIGURATION       88         4.16.6.       SAVE ALARM REPORTS CONFIGURATION       890		4.11.	OPC Filtering Configuration	
4.13.       Clear View		4.12.	Apply Filter Constraints	
4.14.       Print		4.13.	Clear View	
4.15.       Save       82         4.16.       Properties       82         4.16.1.       OPC Options Configuration       83         4.16.2.       OPC ALARM COLOR CONFIGURATION       83         4.16.3.       EMAIL ALARMS CONFIGURATION       83         4.16.4.       CONTROL LAYOUT CONFIGURATION       85         4.16.5.       FILTER CONSTRAINTS CONFIGURATION       88         4.16.6.       Save ALARM REPORTS CONFIGURATION       90		4.14.	Print	
4.16.Properties824.16.1.OPC OPTIONS CONFIGURATION834.16.2.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION844.16.4.CONTROL LAYOUT CONFIGURATION854.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION90		4.15.	Save	
4.16.1.OPC OPTIONS CONFIGURATION834.16.2.OPC ALARM COLOR CONFIGURATION834.16.3.EMAIL ALARMS CONFIGURATION844.16.4.CONTROL LAYOUT CONFIGURATION854.16.5.FILTER CONSTRAINTS CONFIGURATION884.16.6.SAVE ALARM REPORTS CONFIGURATION90		4.16	Properties	
4.16.2.OPC Alarm Color Configuration834.16.3.Email Alarms Configuration844.16.4.Control Layout Configuration854.16.5.Filter Constraints Configuration884.16.6.Save Alarm Reports Configuration90	4.16.	1.	OPC OPTIONS CONFIGURATION	
4.16.3.Email Alarms Configuration844.16.4.Control Layout Configuration854.16.5.Filter Constraints Configuration884.16.6.Save Alarm Reports Configuration90	4.16	2.	OPC ALARM COLOR CONFIGURATION	
4.16.4.CONTROL LAYOUT CONFIGURATION	4.16.	3.	EMAIL ALARMS CONFIGURATION	
4.16.5.       FILTER CONSTRAINTS CONFIGURATION       88         4.16.6.       SAVE ALARM REPORTS CONFIGURATION       90	4.16	4.		85
4.16.6. SAVE ALARM REPORTS CONFIGURATION			CUNTRUL LAYOUT CUNFIGURATION	
	4.16	5.	FILTER CONSTRAINTS CONFIGURATION	

	4.17.	Logging	
	4.18.	About Box Dialog	
2.	REGISTERING THE OPC AE LOGGER .NET ACTIVEX9		
3.	UNREC	GISTERING THE OPC AE LOGGER .NET ACTIVEX	95
4.	DEPLO	YING THE .NET AE LOGGER IN MICROSOFT VISUAL BASIC 6.0	97
	4.1.	Create a Standard EXE	
	4.2.	Add the OPC AE Net Logger reference	
	4.1.	Add the OPC AE Net Logger Component to the Toolbar	
USII	NG OPC I	EVENT CLIENT ACTIVEX	
1.	IOEVE	NTCLIENTCTRL	
	1.1.	Logging	
	1.2.	GetLocalOPCEventServers	
	1.3.	GetOPCEventServers	
	1.4.	CreateServer	
	1.5.	Servers	
	1.6.	AboutBox	
2.	IOEVE	NTSERVER	
	2.1.	ConnectToServer	
	2.2.	Disconnect	
	2.3.	CreateEventSubscription	
	2.4.	QueryEventCategories	
	2.5.	CategoryID	
	2.6.	CategoryDescription	
	2.7.	QueryEventAttributes	
	2.8.	AckCondition	
	2.9.	EnableConditionByArea	
	2.10.	EnableConditionBySource	
	2.11.	DisableConditionByArea	
	2.12.	DisableConditionBySource	
	2.13.	QueryConditionNames	
	2.14.	ConditionName	
	2.15.	QuerySubConditionNames	
	2.16.	SubConditionName	
	2.17.	QuerySourceConditionNames	
	2.18.	SourceConditionName	
	2.19. 2.20	GelConalionState	
2			123
5.			
	3.1. 2 2	GetSubscriptionState	
	3.2. 2 2	Seisuustiipilulislule Pafrash	124 124
	5.5. 21	nejiesii CancalRafrash	
	3.4. 2 5	Activato	
	3.5.	Deactivate	
	37	SelectAllAttributesForAllCataories	
	3.8	GetReturnedEventAttributes	
	0.01		

integration objects



	3.9.	SelectReturnedEventAttributes	
	3.10.	GetFilter	128
	3.11.	ApplyFilter	129
	3.12.	AddFilterSource	
	3.13.	AddFilterArea	
	3.14.	AddFilterCategory	
	3.15.	RemoveFilterSource	
	3.16.	RemoveFilterArea	
	3.17.	RemoveFilterCategory	
	3.18.	ClearFilterSources	
	3.19.	ClearFilterAreas	
	3.20.	ClearFilterCategories	
4.	IOEVEN	ITCONDITIONSTATE	133
5.	IOEVEN	іт	134
6.	IOEVEN	ITSERVERSTATUS	136
7.	IOEVEN	ITBROWSER	137
	7.1.	ShowBrowserDialog	
8.	IOEVEN	ITATTRIBUTES	138
9.	ACTIVE	X DEFINED ENUMERATIONS	138
	9.1.	IO OPCAE SERVERSTATE CONSTANTS Enumeration	
	9.2.	IO OPCAE FILTER CONSTANTS Enumeration	
	9.3.	IO OPCAE EVENTTYPES CONSTANTS Enumeration	
	9.4.	IO_OPCAE_CONDITIONSTATE_CONSTANTS Enumeration	
	9.5.	IO_OPCAE_CHANGE_CONSTANTS Enumeration	
USIN	IG OPC E	VENT CLIENT .NET ACTIVEX	140
10.	IOEVEN	ITCLIENTDOTNETCTRL	140
	10.1.	Logging	
	10.2.	GetLocalOPCEventServers	
	10.3.	GetOPCEventServers	
	10.4.	CreateServer	
	10.5.	GetServer	
	10.6.	AboutBox	
	10.7.	InitializeEventCallback	
11.	OPCAE	SERVER	144
	11.1.	ConnectToOPCServer	
	11.2.	DisconnectfromServer	
	11.3.	CreateEventSubscription	
	11.4.	QueryEventCategories	
	11.5.	GetCategoryID	
	11.6.	GetCategoryDescription	
	11.7.	QueryEventAttributes	
	11.8.	AckCondition	
	11.9.	AckConditionActiveFileTimeAsString	
	1.1.	EnableConditionByArea	
	12	EnableConditionBySource	1/0



	1.3.	DisableConditionByArea	
	1.4.	DisableConditionBySource	
	1.5.	QueryConditionNames	
	1.6.	GetConditionName	
	1.7.	QuerySubConditionNames	
	1.8.	GetSubConditionName	
	1.9.	QuerySourceConditionNames	
	1.10.	GetSourceConditionName	
	1.11.	GetConditionState	
	1.12.	GetCanFilterByEvent	
	1.13.	GetCanFilterBySeverity	
	1.14.	GetCanFilterByCategory	
	1.15.	GetCanFilterByArea	
	1.16.	GetCanFilterBySource	
	1.17.	CreateBrowser	
	1.18.	GetCategoryDescriptionfromID	
	1.19.	GetCategoryDescriptionfromID	
	1.20.	GetCategoryIDfromDescription	
2.	OPCEV	/ENTSUBSCRIPTION	
	21	GetSubscriptionState	156
	2.2.	SetSubscriptionState	
	2.3.	Refresh	
	2.4.	CancelRefresh	
	2.5.	Activate	
	2.6.	Deactivate	
	2.7.	SelectAllAttributesForAllCataories	
	2.8.	GetReturnedEventAttributes	
	2.9.	SelectReturnedEventAttributes	
	2.10.	GetFilter	
	2.11.	ApplyFilter	
	2.12.	AddFilterSource	
	2.13.	AddFilterArea	
	2.14.	AddFilterCategory	
	2.15.	RemoveFilterSource	
	2.16.	RemoveFilterArea	
	2.17.	RemoveFilterCategory	
	2.18.	ClearFilterSources	
	2.19.	ClearFilterAreas	
	2.20.	ClearFilterCategories	
3.	ОРССС	ONDITIONSTATE	
	3.1.	GetQuality	
	3.2.	GetState	
	3.3.	GetActiveSubConditionDefinition	
	3.4.	GetActiveSubConditionDescription	
	3.5.	GetActiveSubConditionSeverity	
	3.6.	GetSubConditionLastActiveTime	
	3.7.	GetConditionLastActiveTime	
	3.8.	GetConditionLastInactiveTime	
	3.9.	GetStateString	
	3.10.	GetSubConditionsCount	



	3.11.	GetSubConditionDefinitions	
	3.12.	GetSubConditionDescriptions	
	3.13.	GetSubConditionSeverities	
	3.14.	GetSubConditionNames	
	3.15.	GetQualityString	
	3.16.	GetSubConditionLastActiveTimeAsString	
	3.17.	GetConditionLastActiveTimeAsString	
	3.18.	GetSubConditionLastInactiveTimeAsString	
	3.19.	GetLastAckTimeAsString	
4.	EVENT	STRUCT	167
	4.1.	GetEventAttributes	
	4.2.	GetEventAttributesAsString	
	4.3.	GetStrQuality	
	4.4.	GetEventTimeAsString	
	4.5.	GetActiveTimeAsString	
5.	OPCSE	RVERSTATUS	170
	4.1.	GetServerStatusString	
	4.2.	GetStartTimeString	
	4.3.	GetLastUpdateString	
	4.4.	GetCurrentTimeString	
6.	OPCE\	/ENTBROWSER	
	6.1.	ShowBrowserDialog	
7.	OPCE\	/ENTATTRIBUTES	172
	7.1.	GetAttributeIDs	
	7.2.	GetAttributeDescriptions	
	7.3.	GetAttributeVarTypes	
	7.4.	GetVarTypeString	
8.	OPCE\	/ENTFILTER	173
	8.1.	SetEventCategories	
	8.2.	SetFilterAreas	
	8.3.	GetFilterAreas	
	8.4.	GetEventCategories	
	8.5.	GetFilterSources	
	8.6.	SetFilterSources	
9.	OPCE\	/ENTCONSTANTS	175
	9.1.	OPC FILTER CONSTANTS	
	9.2.	OPC EVENT TYPES CONSTANTS	
	9.3.	OPC CONDITION STATE CONSTANTS	
	9.4.	OPC CHANGE CONSTANTS	
5.	DEPLO	YING THE EVENT CLIENT .NET ACTIVEX IN MICROSOFT VISUAL BASIC 6.0	176
	5.1.	Create a Standard EXE	
	5.2.	Add the OPC Event Client .Net reference	
	5.3.	Add the OPC Event Client Net ActiveX Component to the Toolbar	
CON	NFIGURI	NG DCOM	



1.	CLIENT SIDE DCOM CONFIGURATION	184
10.	SERVER SIDE DCOM CONFIGURATION	190



## **TABLE OF FIGURES**

Figure 1 : Welcome Setup Wizard	. 20
Figure 2: License Agreement Window	. 21
Figure 3: Customer Information	. 22
Figure 4: Select Destination Folder	. 23
Figure 5: Confirm Install	. 23
Figure 6: Installation Progress Window	. 24
Figure 7: Installation Complete Window	. 24
Figure 8: Start Menu	. 25
Figure 9: Start Menu – Uninstaller Shortcut	. 26
Figure 10: Windows 10 Startup Menu - Uninstall ShortcutInstalled Files	. 26
Figure 11: Integration Objects' OPC AE ActiveX Installation Directory	. 27
Figure 12: Dependencies	. 27
Figure 13: Integration Objects' OPC AE ActiveX	. 30
Figure 14: Integration Objects' Alarms Logger Inserted into the VB Form	. 31
Figure 15: Integration Objects' Alarms Logger Retrieving Real-Time Alarms	. 31
Figure 16: Alarms Logger OPC Options	. 33
Figure 17: Entering AE Server Settings	. 34
Figure 18: Configure Email Notifications	. 36
Figure 19: Alarms Logger Layout	. 37
Figure 20: Pick Up Columns to Display in the Logger	. 38
Figure 21: Enabling Reordering Columns	. 38
Figure 22: Drag-and-Drops Column Headers to Reorder Columns	. 38
Figure 23: Customizing Data	. 39
Figure 24: Filter Constraints	. 39
Figure 25: Add Constraint Dialog	. 40
Figure 26: Select the Constraint Field	. 40
Figure 27: Select the Constraint Operator	. 40
Figure 28: Constraint Using Wildcards	. 41
Figure 29: Apply Changes Notification Message Box	. 41
Figure 30: Save Alarms Reports	. 42
Figure 31: Select Reports Folder	. 43
Figure 32: Log Configuration Dialog	. 45
Figure 33: Choosing the Log Level	. 45
Figure 34: Context Menu	. 45
Figure 35 View Menu Item	. 46
Figure 36: "Connect" from Context Menu	. 47
Figure 37: Acknowledging Alarms	. 48
Figure 38: Server Status	. 49
Figure 39: Event Subscription Status	. 50
Figure 40: Select Specific Server Returned Attributes	. 51
Figure 41: Get Specific Server Returned Attributes	. 52
Figure 42: Enable/Disable Condition Menu Item	. 52
Figure 43: Enable Condition by Source	. 53
Figure 44: Enable Condition by Area	. 53



Figure 45: Enable Condition by Area	54
Figure 46: Enable Condition by Area	54
Figure 47: Available Filters	55
Figure 48: Configure OPC Filtering Alarms	56
Figure 49: Add Areas	57
Figure 50: Add Sources	57
Figure 51: Filter Constraints Window at Runtime	58
Figure 52: About Box	59
Figure 53: Context Menu	64
Figure 54: View Menu Item	65
Figure 55: Auto Load Configuration Menu Item	65
Figure 56: Load Configuration Menu Item	66
Figure 57: Clean Configuration Menu Item	66
Figure 58: Connect From Context Menu	67
Figure 59: Disconnect Menu Item	67
Figure 60: Acknowledge Menu Item	67
Figure 61: Acknowledging Alarms	68
Figure 62: Show Acknowledge Dialog Menu Item	68
Figure 63: Show Acknowledge Result Menu Item	69
Figure 64: Server Status	70
Figure 65: Event Subscription State	70
Figure 66: Select All Attributes for All Categories Menu Item	71
Figure 67: Select Specific Server Returned Attributes	71
Figure 68: Get Specific Server Returned Attributes	72
Figure 69: Enable/Disable Condition Menu Item	72
Figure 70: Enable Condition by Source	73
Figure 71: Enable Condition by Area	73
Figure 72: Disable Condition by Area	74
Figure 73: Disable Condition by Source	74
Figure 74: Available Filters	75
Figure 75: Configure OPC Filtering Alarms	76
Figure 76: Add Areas	77
Figure 77: Add Sources	78
Figure 78: Filter Constraints Window	79
Figure 79: Clear View Menu Item	79
Figure 80: Print Configuration	79
Figure 81: Page Setup Dialog Box	80
Figure 82: Preview Print Dialog Box	81
Figure 83: Fit to Page Check Box	81
Figure 84: Choose Printer	82
Figure 85: Save Item Menu	82
Figure 86: Properties Menu Item	82
Figure 87: Alarms Logger OPC Options	83
Figure 88: Configure Alarms Colors	84
Figure 89: Configure Email Notifications	85
Figure 90: Alarms Logger Layout	86
Figure 91:Pick Up Columns to Display	87
Figure 92: Enabling Reordering Columns	87
Figure 93: Drag-and-Drops Column Headers to Reorder Columns	88
Figure 94: Customizing Data Format	88



Figure 95: Filter Constraints	. 89
Figure 96: Add Constraint Dialog	. 89
Figure 97: Select the Constraint Field	. 90
Figure 98: Select the Constraint Operator	. 90
Figure 99: Warning Message Box	. 90
Figure 100: Save Alarms Reports	. 91
Figure 101: Select Reports Folder	. 92
Figure 102: Log File Setting	. 93
Figure 103: About Box	. 94
Figure 104: Registration Command Prompt	. 95
Figure 105: Un-registration Command Prompt	. 96
Figure 106: Create a VB6 Standard EXE	. 97
Figure 107: Select Projet Reference	. 98
Figure 108: Browse the OPC AE Net Logger Path	. 99
Figure 109: Select the type library (.tlb) file	. 99
Figure 110: Check the OPC AE Net Logger Reference	100
Figure 111: Select Projet Components	101
Figure 112: Select the OPC AE Net Logger Component	102
Figure 113: Add the OPC AE Logger Component to the Form	103
Figure 114: Create a VB6 Standard EXE	177
Figure 115: Select Projet Reference	178
Figure 116: Browse the OPC AE Net Logger Path	179
Figure 117: Select the type library (.tlb) file	179
Figure 118: Check the OPC Event Client Net ActiveX Reference	180
Figure 119: Select Projet Components	181
Figure 120: Select the OPC Event Net ActiveX Component	182
Figure 121: Add the OPC Event Client Net ActiveX Component to the Form	183
Figure 122:Distributed COM Configuration	185
Figure 123: DCOM Default Properties	186
Figure 124:DCOM Default Security Tab	187
Figure 125 : DCOM Default Protocols	188
Figure 126: General properties of the selected OPC server	189
Figure 127 :DCOM Server Configuration	191
Figure 128:Location configuration of the selected OPC server	192
Figure 129: Configuration of DCOM security properties	193



## **LIST OF TABLES**

Table 1: Installed Directories	
Table 2: The ActiveX Columns	
Table 3: Configuration File Properties	44
Table 4: Acknowledgment Options	48
Table 5: The ActiveX Columns	87
Table 6: Log File Properties	
Table 7: Configuration File Properties	
Table 8: Configuration File Properties	



## PREFACE

## About this Guide

This guide:

- Explains how to install, configure and use the Integration Objects' OPC AE ActiveX, also referred to as OPC AE Controls.
- Describes the system requirements for this release.

### Audience

This manual is intended for programmers who wish to use Integration Objects' OPC AE ActiveX. It assumes that readers are familiar with Microsoft ActiveX technologies and have experience with the programming language (for example, Visual Basic) needed to create their ActiveX application.

It also assumes that you have an idea about OPC (OLE for Process Control) and OPC Alarms and Events specification.

### **Related Documentation**

**OPC** Foundation

As you use this user guide, you may also find the following specification useful: A&E Custom Interface Standard Version 1.10.

#### ActiveX Documentation

For more information about using ActiveX, refer to your ActiveX container documentation (for example, the Visual Basic documentation) or see Microsoft's web page: http://www.microsoft.com



## **CUSTOMER SUPPORT SERVICES**

Phone	Email
Americas:	Support:
+1 713 609 9208	customerservice@integrationobjects.com
	Sales:
Europe-Africa-Middle East	sales@integrationobjects.com
+216 71 195 360	Online:
	www.integrationobjects.com



## **GETTING STARTED**

This chapter describes the ActiveX Control, the basic steps for using it, and installation information.

#### 1. Overview

ActiveX control is one element of the Microsoft's ActiveX object-based technology. It is a piece of software that can be used with any application or programming environment defined as an ActiveX container. The container provides a framework in which the control operates, and collectively, with other ActiveX components, container and control provide whatever functions are required of the ActiveX software design. This requirement may be to interact with other ActiveX controls or containers or to run as a complete application.

Integration Objects' OPC AE ActiveX provides access to the basic features of the OPC AE API. Using these controls, you can develop an AE application that allows users to view real time alarms received by an OPC AE Server.

If you are a developer creating OPC client applications, this toolkit was designed for you. This customizable control can be embedded into any Web page or Windows® application.

The control can be used with any ActiveX container, including applications such as Microsoft Access or programming environments such as Visual Basic.

### 2. Software Components

The software contains the following software components and documents:

- Alarm Logger Control: this control allows users to display current alarms and events in a list view.
- *Alarm Logger .Net Control*: this control allows users to display current alarms and events in a list view. This is the preferred logger control to be deployed with .Net based ActiveX containers.
- **AE Servers ListBox**: This control offers the possibility to retrieve OPC AE Servers in a list box.
- **AE Servers ComboBox**: the same as the previous control, the AE Servers ComboBox is a combo box control, which retrieves the OPC AE Servers registered locally or remotely.



- **Event Client Component**: This component allows VB and OLE containers to quickly and easily access data from any OPC A&E Server. Methods and properties exist within the control to allow the developed application to connect to the AE Server on the local or on the remote machine and then to access OPC AE Server interfaces.
- Event Client .Net Component: This component allows VB and OLE containers to quickly and easily access data from any OPC A&E Server. Methods and properties exist within the control to allow the developed application to connect to the AE Server on the local or on the remote machine and then to access OPC AE Server interfaces. This is the preferred event client control to be deployed with .Net based ActiveX containers.
- Sample application with source code is provided to demonstrate the capabilities of the toolkit.
- The software User Guide (this document).
- The software Quick User Guide.

### **3. Features**

This section contains a brief description of each of the components of Integration Objects' OPC AE ActiveX software.

Integration Objects' OPC Alarm Logger

- The *Alarm Logger Control* delivers a great GUI enhancement to VB applications in other words, the ability to dock or float regular VB controls that connect and display data without any line of the code.
- Auto-discovery of all OPC AE servers available on the network.
- Connection to an AE OPC Server locally or remotely.
- Alarm filtering to view only certain types of incoming alarms.
- By adding customized filter constraints for incoming alarms, users can simply define filters using wildcards such as (?, \*).
- Acknowledge events operator.
- Special text and background colors for alarms.
- Notifications of particular alarms sent to administrator by email.
- Archiving alarms automatically in CSV files.
- The user can follow the operations handled by the control by saving log entries as files.
- Enabling OLE Automation containers (such as VB, VBA, Internet Explorer) to view incoming alarms.
- Complete scrollable display.
- Change column information to be viewed.
- Configurable message filters to view only certain types of messages.
- Display server information on demand.

Integration Objects' OPC AE Alarm Logger .Net ActiveX



- The *Alarm Logger .Net Control* delivers a great GUI enhancement to VB applications in other words, the ability to dock or float regular VB controls that connect and display data without any line of the code.
- Auto-discovery of all OPC AE servers available on the network.
- Connection to an AE OPC Server locally or remotely.
- Alarm filtering to view only certain types of incoming alarms.
- By adding customized filter constraints for incoming alarms, users can simply define filters using wildcards such as (?, \*).
- Acknowledge events operator.
- Special text and background colors for alarms.
- Notifications of particular alarms sent to administrator by email.
- Archiving alarms automatically in CSV files.
- The user can follow the operations handled by the control by saving log entries as files.
- Enabling OLE Automation containers (such as Power Builder, Proficy iFIX, Proficy Cimplicity) to view incoming alarms.
- Complete scrollable display.
- Change column information to be viewed.
- Configurable message filters to view only certain types of messages.
- Display server information on demand.
- Print the incoming alarms

Integration Objects' OPC Event Client .Net ActiveX

- Auto-discovery of all OPC AE servers available on the network in a ListBox format.
- Connection to OPC AE servers locally and remotely
- Enabling OLE Automation containers (such as VB, VBA, Internet Explorer) to view incoming alarms.
- Acknowledge events operator.
- Display server information on demand.
- Alarm filtering to view only certain types of incoming alarms.
- Complete scrollable display.

Integration Objects' OPC Event Client ActiveX

- Auto-discovery of all OPC AE servers available on the network in a ListBox format.
- Connection to OPC AE servers locally and remotely
- Enabling OLE Automation containers (such as Power Builder, Proficy iFIX, Proficy Cimplicity) to view incoming alarms.
- Acknowledge events operator.
- Display server information on demand.
- Alarm filtering to view only certain types of incoming alarms.
- Complete scrollable display.

Integration Objects' Event Servers ListBox



• Auto-discovery of all OPC AE servers available on the network in a ListBox format.

Integration Objects' Event Servers ComboBox

• Auto-discovery of all OPC AE servers available on the network in a ComboBox format.

### 4. Software Requirements

Microsoft .Net Framework version 2 Redistributable Package or a more recent version is required for a proper installation of the OPC AE ActiveX.

To develop applications using Integration Objects' OPC AE ActiveX, you will require the following:

- Operating System :
  - Windows XP
  - Windows Server 2003
  - Windows 7
  - Windows 8 and 8.1
  - Windows 10
  - Windows Server 2008
  - Windows Server 2012
  - Windows Server 2016
- ActiveX/COM aware programming environment, such as Visual Basic®, Delphi<sup>™</sup>, etc. (only on the development platform).

Microsoft Visual VB 6.0 (with Service Pack 4) or higher may be required as well, for users to develop their own applications.

### 5. OPC Compatibility

- OPC A&E specification version 1.02
- OPC A&E specification version 1.10

### 6. Installing OPC AE ActiveX

The product package consists of several components that need to be loaded in your system. The setup application is designed to place all necessary files into the correct locations. Follow the installation wizard instructions. Click *Next* to proceed from screen to screen in the installation program. You can use the *Back* button to return to earlier screens and change any information that is incorrect. Click *Cancel* to cancel the installation process at any time.

The following section includes the screen shots taken during the installation process.

1. Double-click the *Setup.exe* and the Welcome to the Integration Objects' OPC AE ActiveX Setup Wizard window will appear (see Figure 1).



Integration Objects' OPC AE Act	iveX - InstallShield Wizard	×
	Welcome to the InstallShield Wizard for Integration Objects' OPC AE ActiveX The InstallShield Wizard will install Integration Objects' OPC AE ActiveX on your computer. T continue, click Next.	0
InstallShield	Car	ncel

Figure 1 : Welcome Setup Wizard

- 2. Click the *Next* button to begin the installation process.
- 3. After the welcome window is dismissed, the license window shown below in Figure 2 is displayed. Please read the license agreement and select "I Agree" to proceed with the installation.



Integration Objects' OPC AE Act	tiveX - InstallShield Wizard	x
License Agreement Please read the following licen	se agreement carefully.	
	Integration Objects End-User License & Customer Support and Services Agreement Integration Objects End-User License Agreement PLEASE READ THIS END-USER LICENSE AGREEMENT (the "Agreement") CAREFULLY BEFORE PROCEEDING. THIS AGREEMENT LICENSES THE SOFTWARE TO YOU AND CONTAINS WARRANTY AND LIABILITY DISCLAIMERS. BY INSTALLING THE SOFTWARE, YOU ARE CONFIRMING YOUR ACCEPTANCE OF THE SOFTWARE AND AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT WISH TO DO SO, CLICK "Do not accept" BELOW AND DO NOT INSTALL THE SOFTWARE. 1. Definitions (a) "IO Software" means the software program covered by this Agreement and all related updates supplied by IO. (b) "IO Product" means the 10 Software and the related documentation and content and all related updates supplied by IO. 2. License TIMA THE INFORMANCE IN THE INFORMANCE INFORMANCE INFORMANCE I go not accept the terms of the license agreement I go not accept the terms of the license agreement	
InstallShield	< <u>B</u> ack <u>N</u> ext > Can	icel

Figure 2: License Agreement Window

4. After you agree to the software license, the window shown in Figure 3 is displayed. This window asks you to introduce information about yourself and your company.



Integration Objects' OPC AE Activ	eX - InstallShield Wizard	×
Customer Information Please enter your information.		
	Please enter your name and the name of the company for which you work.	
	User Name:	
	Integration Objects	
InstallShield	( <u>B</u> ack <u>N</u> ext≻ Car	ncel

Figure 3: Customer Information

- 5. A folder selection window appears when you click the next button (see Figure 4).
- 6. By default, all components will be installed under the directory "\Program Files\Integration Objects\Integration Objects' OPC AE ActiveX" unless the user specifies another location during the installation. In most instances, the default folder is the best place for installing the software. If a different directory must be used, click the *Browse* button to search for the directory where you want to install the product and then click the *OK* button.



Integration Objects' OPC AE ActiveX - InstallShield Wizard	x
Choose Destination Location Select folder where setup will install files.	
Setura will install Integration Objects' OPC AE ActiveX in the following folder.         Choose Folder         Please select the installation folder.         Bath:         Nutegration Objects' OPC AE ActiveX         Directories:         Ox DVD Maker         Integration Objects' OPC AE Active         Integration Objects' OPC AE Set         Integration Objects' OPC AE Active         Integration Objects' OPC AE Set         Integration Objects' OPC AE Set         Integration Objects' OPC AE Set         Integration Objects' OPC AE Set	
InstallShield <back next=""> Cance</back>	

Figure 4: Select Destination Folder

7. Click Next and a confirm installation window appears (see Figure 5).



Figure 5: Confirm Install



8. To begin installation, click the *Next* button. An installation progress window will appear (see Figure 6).



Figure 6: Installation Progress Window

9. An Installation Complete window will appear when the product has been successfully installed (see Figure 7).



Figure 7: Installation Complete Window



10. Click the *Finish* button to exit the installation window.

The installation copies all necessary files to the target computer, creates a short-cut icon to place the configuration tool in the Start menu and makes an un-installation entry in the Add/Remove Programs Window in the Control Panel.

Click on Start → Programs → Integration Objects → OPC AE ActiveX



Figure 8: Start Menu

## 7. Uninstalling OPC AE ActiveX

First you should close all applications using the control. Then you uninstall it by using the **Add/Remove Programs** wizard from MS Windows **Control Panel**.

To uninstall the toolkit:

- 1. Make sure there are no running applications using Integration Objects' OPC AE ActiveX.
- 2. Open the Add/Remove Programs control panel.
- 3. Select "Integration Objects' OPC AE ActiveX", and click Change/Remove. The uninstall process completely removes all files that were installed. This does not remove files and folders that were created after the installation.
- 4. After you run the uninstall program, check the installation directory and remove any unnecessary files and sub directories. After uninstalling, there may still be files stored in the installation directory. You should delete these as well.

The software can also be removed using the ""Uninstall OPC AE ActiveX" shortcut available in the start menu.









If you are using Windows 10, Windows Server 2012 or Windows Server 2016 operating systems, the uninstaller needs to be run from the start menu as illustrated below.



Figure 10: Windows 10 Startup Menu - Uninstall Shortcut



#### **Installed Files**



#### Figure 11: Integration Objects' OPC AE ActiveX Installation Directory

When the installation is complete, you should see the following directories installed under the destination folder that you selected:

Directory	Definition
Bin	This directory contains the needed ocx, dll files.
Docs	This directory contains this user guide (.pdf).
DotNet	This directory contains the .Net ActiveX controls dll files
Samples	This directory contains the VB6 samples included within the product.

#### **Table 1: Installed Directories**

### 8. Dependencies



Figure 12: Dependencies

Integration Objects' OPC AE ActiveX requires the Microsoft Foundation Class (MFC) Runtime modules. These modules are usually found in the Windows system directory.



Integration Objects' OPC AE ActiveX dependencies include the following files: mfc70D.dll, msvcp70.dll, msvcr70.dll, mpr.dll, opcdll.dll, WSOCK32.dll.

There may be additional files provided by Windows which are not listed here. If the control fails to register with the above files, you may use the Dependency Walker installed with the Visual Studio to get the dependencies list.



## **USING OPC AE ACTIVEX**

This chapter will explain how to program user applications. A simple application is provided with source code to help users get started using Integration Objects' OPC AE ActiveX. The sample source code will be stored in the "Samples" folder under the product installation location.

You can insert the product ActiveX controls in:

- A VB window (or a window built with another application development tool that supports ActiveX).
- An HTML page.

The first part of this chapter focuses on the steps needed to develop a minimal OPC application using Integration Objects' OPC AE ActiveX with Microsoft Visual Basic 6.0.

#### **1. Start Example**

When starting to write an application, first add the component to your VB project. The following steps show how to incorporate the control into a VB6.0 project.

**Step 1**: Start Visual Basic. Choose *New Project | Standard EXE*. A project named Project1 with a form called Form1 will be displayed.

Step 2: Select *Project* | *Components*. Uncheck "Selected Items Only" box to show all components, and check two components called *Integration Objects' OPC AE Controls Control module* and *Integration Objects' OPC Event Client ActiveX*. If these components are not listed here this means that the product was not correctly installed. Refer to the Install section about how to install and register these components for design and development purposes.

**Step 3**: Once these controls are available in your project, new icons on the left toolbar will appear, and the user can simply drag and drop any of these icons to the Form. Four controls are available: *Integration Objects' OPC Alarms Logger*, *Integration Objects' Event Servers ListBox*, *Integration Objects' Event Servers ComboBox* and *Integration Objects' OPC Event Client ActiveX*.







Figure 13: Integration Objects' OPC AE ActiveX

Integration Objects' Servers ListBox and Integration Objects' Servers ComboBox are simple controls which allow users to get locally or remotely registered AE Servers into ListBox or ComboBox.



_	
G	enEvent.1
10	CONICS.AlarmServer.1
Ir	ntegrationObjects.OPCAE.vMonitor
N	atrikon.0PC.Simulation.1
s	ofting.OPCToolboxDemo ServerAl
	5 _

*Integration Objects' Alarms Logger* allows the user to view current alarms and handles alarm acknowledgement. Below is a VB Form including *Integration Objects' Alarms Logger*. This control represents a list view. Users can modify the layout of the information, including order, and displayed data.





egration Object	s' Sample Using AE	Logger						
n								
Event Quality	•							
Event Type								
Message								
Event Time								
Event Time	Active Time	Source	Condition	Sub Condition	Message	NewState	Severity	

#### Figure 14: Integration Objects' Alarms Logger Inserted into the VB Form

Integration Object	s' Sample Using AE	Logger	-			2. 2		
Event Quality	•							
Event Type	Condition							
Message	HI Alarm							
Event Time	25/09/2012 16:05:	36						
Event Time	Active Time	Source	Condition	Sub Condition	Message	NewState	Severity	Q A
25/09/2012 16:	25/09/2012 16:	FIC1001	PVLEVEL	LO	LO Alarm	ACTIVE ENABL	300	Good.N
25/09/2012 16:	25/09/2012 16:	FIC1001	PVLEVEL		Condition Normal	ENABLED	300	Good.N
25/09/2012 16:	25/09/2012 16:	FIC1001	PVLEVEL	LOLO	LOLO Alarm	ACTIVE ENABL	100	Good.N
25/09/2012 16:	25/09/2012 16:	FIC1002	DEVIATION	DEVIATION	DEVIATION Alarm	ACTIVE ENABL	500	Good.N
25/09/2012 16:	25/09/2012 16:	FIC1002	DEVIATION		Condition Normal	ENABLED	500	Good.N
25/09/2012 16:	25/09/2012 16:	FIC1003	PVLEVEL		Condition Normal	ENABLED	300	Good.N
25/09/2012 16:	25/09/2012 16:	FIC1003	PVLEVEL	HIHI	HIHI Alarm	ACTIVE ENABL	900	Good.N
	25/09/2012 16:	FIC1003	PVLEVEL		Condition Normal	ENABLED	900	Good.N
	25/09/2012 16:	FIC1003	PVLEVEL	HI	HI Alarm	ACTIVE ENABL	700	Good.N
	25/09/2012 16:	FICT003	PVLEVEL	10	Condition Normal	ENABLED	700	Good.N
	25/03/2012 16:	FICT003	PVLEVEL	LU	LU Alarm	AUTIVE ENABL	300	Good.N
	25/09/2012 16:	FIC1003	PVLEVEL	1010	Londition Normal	ENABLED	300	Good.N
	25/05/2012 16:	FICT003	DEVIATION	DEVIATION	DEVIATION Alarm	ACTIVE ENABL	300	Good N
	20/00/2012 10	FIC1004	DEVIATION	DEVIATION	Condition Margal	ENABLED	500	Good N
25/09/2012 16	20/03/2012 16	Sustem Event	DEVIATION		Simple quest	ENABLED	200	G000.N
25/09/2012 16:		Tracking EVENT			Tracking event		500	
25/09/2012 16	25/09/2012 16	FIC1001	PVI EVEI		Condition Normal	ENABLED	100	Good N
25/09/2012 18	25/09/2012 18	FIC1001	PVLEVEL	нн	HIHI Alarm	ACTIVE ENABL	900	Good N
25/09/2012 16	25/09/2012 16	FIC1001	EVI EVEL		Condition Normal	ENABLED	900	Good N
25/09/2012 16:	25/09/2012 16:	FIC1001	PVLEVEL	HI	HI Alarm	ACTIVE ENABL	700	Good.N
								- F

Figure 15: Integration Objects' Alarms Logger Retrieving Real-Time Alarms



*Integration Objects' OPC Event Client ActiveX* allows the user to implement an AE Client using a set of methods and properties.



## **CONFIGURING THE AE LOGGER**

## **1. Design-Time Configuration**

Integration Objects' Alarms Logger includes property pages that enable you to easily configure the product to connect to an OPC AE Server and to retrieve incoming Alarms and Events. To view or modify the Alarm Logger properties, right click on the control and choose Properties. The dialog in Figure 14 will appear.

Below are screenshots of the control properties dialog and a description of its use.

#### **1.1. OPC Options Configuration**

Alarms Logger OPC Options Use this Property Page to define the OPC AE Address.  Alarms Logger Layout Filter Constraints OPC Configure Alarms Colors Choose the OPC AE Server	Server Name and the A Save Alams Reports Email Alams	NE Server OK Cancel
OPC AE Server : IntegrationObjects.OPC.AE.Knet.Server Server Address : DEV_04 DEV_04 Microsoft Windows Network Web Client Network DEV_04 IntegrationObjects.OPC.AE.Knet.Server IntegrationObjects.OPC.AE.Knet.Server IntegrationObjects.OPC.AE.Knet.Server Matrikon.OPC.Simulation Watrikon.OPC.Simulation.1 Softing.OPCToolboxDemo_ServerAE.1		
Auto Launch the OPC AE Server		

Figure 16: Alarms Logger OPC Options



Figure 14 represents the first tab sheet of the properties dialog, which is used to define the OPC AE Server and the AE Server location. The tree-view in the tab allows the user to browse the server he wants.

To pick up an AE server, you can browse the registered AE servers by simply double clicking on a machine node. You can also enter the AE server name and the server address in the edit boxes (Figure 15).

OPC AE Server :	IntegrationObjects.OPC.AE.Knet.Server
Server Address :	DEV_04

#### Figure 17: Entering AE Server Settings

#### **1.2. OPC Alarm Color Configuration**

Integration Objects' Alarm Logger offers several options to configure GUI to be efficient and to respect the users' preferences. To change the configuration, use the "Configure ALARMS Colors" tab sheet in the main properties window.

The next tab sheet gives users the possibility to adjust background and text colors for alarms. Colors are an important piece of information for the alarm display.

You may also choose to enable blinking of active and unacknowledged alarms by clicking on the "enable blinking" check box.



Integration Objects Alarms Logger Properties	X
Alarms Logger OPC Alarms Colors Click to adjust Backround and Text colors for	· Alarms.
Alarms Logger Layout Filter Constraints OPC Configure Alarms Colors	Save Alarms Reports OK
Colors       Backround Color       T         Active Alarm Not Acknowledged	Cancel

Figure : Enable Blinking

#### **1.3. Email Alarms Configuration**

You can configure the application to send emails when an event or an alarm occurs. The user must provide the destination email address and SMTP server.

Integration Objects' Alarm Logger has a tab sheet used to configure email notifications. When an alarm occurs, the application using the control automatically sends detailed notifications to the administrator by e-mail. The user can specify low or high severity for received alarms and events.

To enable mail notification, you should check the "enable mail notification" check box. If you choose to select the "send only critical alarms" option, you will receive only the incoming alarms that meet the severity level you define.



Integration Objects Alarms Logger Properties	×
Alarms Logger OPC Alarms Notification Mails Use this page to allow the Alarms Logger to send emails to specific an an alarm occurs.	ddresses when
Alarms Logger Layout       Filter Constraints       Save Alarms Reports         OPC       Configure Alarms Colors       Email Alarms         Image: Enable mail notification       Mail Server and sender e-mail address       Email Alarms         Smtp Server       Smtpserver.com       Sender mail address       Sender mail address         Sender mail address       sales@integrationobjects.com       User Name         Password	OK Cancel
Name       Customer Service         Email       customerservice@integrationobjects.com         Image: Send only critical alarms       Image: Send only critical alarms	
High Severity 1000	

Figure 18: Configure Email Notifications

#### **1.4. Control Layout Configuration**

The Control layout tab is used to select the columns you want to appear on your control panel. The following dialog box allows users to view custom information, to pick up columns you want to display in the control panel, and to choose displayed format data to access further information.


Figure 19: Alarms Logger Layout

This ActiveX control includes the following columns:

Column Heading	Description
Event Time	The time that the event was generated.
Active Time	The time that the event was active.
Source	The source item that caused the event to occur.
Condition	The condition name that caused the event to occur.
Sub Condition	The detailed cause of the event.
Message	The alarm message.
Event type	OPC specific event type (simple, tracking, condition).
Event category	One of the OPC event categories.
Quality	An indicator to the reliability of the event.



Actor ID	The name of the user that acknowledged the event.
Ack Comment	The comments entered by the user when he acknowledges the event.
Ack required	Indicates if the event requires acknowledgment.
Severity	The severity of an alarm.
Ack Time	The time when the user acknowledged the alarm.
Cookie	
NewState	The type of alarm (active, acknowledged).
Attributes	Vendor specific server attributes.

#### Table 2: The ActiveX Columns

If the user wants to pick up only some screen fields to display in the control, simply uncheck the unwanted columns.

All columns are checked by default except the "Attributes" column which the user may mark if so desired.

- Screen Fields		
	Asla Dava	
Event Time	Ack time	Message
Active Time	🗹 Quality	Condition
<ul> <li>Event Category</li> </ul>	Source	SubCondition
Event Type	Comments	✓ New State
Severity	Cookie	Attributes
ActorID	Ack Required	

Figure 20: Pick Up Columns to Display in the Logger

This Layout dialog allows users to define layout options for the control.

The user can choose the ability to drag-and-drop column headers to reorder columns in the control. By default, drag-and-drop reordering of columns is enabled (Figure 20).

- Layout options	
🗹 Hide Column Headers	Allow column reordering
Show grid lines	

#### Figure 21: Enabling Reordering Columns

To move a column, click the column header once to select it. Then click and drag the column header to a new location (Figure 21).

e Time 🔡	Condition	Source Source	Sub (
005 10:	between	computer.clock.t	bet
601 01:		computer.clock.t	
601 01:		computer.clock.t	

Figure 22: Drag-and-Drops Column Headers to Reorder Columns



We've decided to add a new preference on the layout page which allows you to select the time format for some screen field controls.

Along with choosing a custom date/time format, the user may choose to display date and time or just time for the three fields: **Event Time**, **Active Time** and **Ack Time**.

Format Data				
Event Time	Date and Time	- Ack	Date and Time	•
Active Time	Date and Time		[Description]	•

Figure 23: Customizing Data

# **1.5.** Filter Constraints Configuration

The Filter constraints tab is used to set criteria for including or excluding specific alarms in the list view.

Integration Objects Alarms Logger Properties	<b>X</b>		
Filter Constraints Use this page to add make configurable filter constraints to view only of alarms. You have the possibility to add constraints using wildcards.	y certain types		
OPC Configure Alarms Colors Email Alarms Alarms Logger Layout Filter Constraints Save Alarms Reports	ок		
Enable Filter Constraints     Constraints	Cancel		
Set/Remove check mark to enable/disable filter Boolean Operator Column Name Operator Value			
AND Event Time = 25/09/2012 15			
Add Constraint     Enable Constraint     Disable Constraint     Enable All       Disable All     Remove     Apply			

Figure 24: Filter Constraints

To define a filter constraint, click the *Add Constraint* button and the dialog (Figure 24) will appear. You must select the filter column name, the operator (=, #, <, ...) and the value.



Add Constraint
New Constraint
⊙ AND O OR
Column Name Operator Value
Event Time 🗨 = 💌 25/09/2012 15:31:20 💌
Expression
AND (EVENTTIME = 25/09/2012 15:31:20)
OK Cancel





Figure 26: Select the Constraint Field



Figure 27: Select the Constraint Operator

In some kind of filter constraints, the user has the possibility to enter string values using wildcards (?, \*, ...).



Add Constraint
New Constraint
⊙ AND O OR
Column Name Operator Value
Message  =  *Between?135?*
if you select the "LIKE" operator, you have the possibility to enter string with wildcards.
Expression
AND (MESSAGE = "*Between?135?*")
OK Cancel

Figure 28: Constraint Using Wildcards

Before moving to another tab, you should click the *Apply* button to save modifications. If you did not do so, the following message box will appear.

	Section 2.	x
You have to Apply changes in the filter co changes.	nstraints page. Click on	"OK" to save
	ОК	Annuler

Figure 29: Apply Changes Notification Message Box

## **1.6.** Save Alarm Reports Configuration

Integration Objects' Alarms Logger allows the user to save the retrieved alarms into report files. To do that, the user has the possibility to define how frequently he wants the save to be performed (each hour or after a certain number of hours).

By default, alarms are saved each hour into the directory where the application deploying the control is located. If you are not pleased with this location, you can select another directory. Click *Browse Path* to select the folder where you want to save the alarms. When saving the alarms report, the toolkit creates a file in the chosen folder which is entitled:

"AlarmsReports{DAY]\_{MONTH]\_[YEAR]\_{HOUR]\_{MIN]\_{SECONDS]" with the "csv" extension.



Use this page to configure the way you save alarms into files.	
OPC Configure Alarms Colors Email Alarms OK	
Alams Logger Layout       Filter Constraints       Save Alams Reports         Save Reports       Save alams each       1       Hours         Browse file path       File Path       C:\Program Files\Integration Objects       Browse Path         Alams are saved automatically to files untitled as the following :       AlammReports25_09_12_15_35_01.csv       Cancel         Logging       Trace Level       Trace only errors       ▼         ✓       Archive Last Log       ✓	

Figure 30: Save Alarms Reports





Figure 31: Select Reports Folder

# 1.7. Logging

Integration Objects' Alarms Logger produces a log file named "IOAEOCX\_LogEvent.LOG" that records errors and debugging information in design time. It also produces a second file named "IOAEOCX\_LogEvent1.LOG" that records errors and debugging information at runtime. If difficulties occur with the application, the log file can be extremely valuable for troubleshooting. When operations are running normally, the control will log very little information.

These log files are generated by default in the bin folder under the installation folder. This path can be modified by the user using the "Configuration.ini" configuration file incorporated by the toolkit which includes several logging parameters and a parameter used to define the folder where the application saves retrieved alarms. These parameters all have default settings and can be changed at start-up by editing the configuration file, or by simply changing the information in the control properties window.



To change this file:

- 1. Open Configuration.ini in a text editor.
- 2. Edit any of the parameters listed in the following tables:

Log Setting	Description	Default Value
LogFileMaxSize	The maximum log file size, in bytes. Once this size is reached during run-time, the log file is overwritten.	1048576*2 ~ 2 Mb (MegaByte)
TraceLevel	<ul> <li>The trace level is a value telling what type of information to log:</li> <li>0: Only errors messages are logged.</li> <li>1: Some extra information.</li> <li>2: Debugging information is logged.</li> <li>3: Detailed debugging information.</li> <li>The higher the trace level, the more information is recorded. We recommend you to use level 0 for a better performance of the client application.</li> </ul>	0
ArchiveLastLog	TRUE: Old file is copied to an intermediate file with incremental extension, before being overwritten. FALSE: Any pre-existing log file is erased and overwritten at start-up.	FALSE
ReportsFolder	The path where the alarms report will be saved	C:
LogFilePath	The path where the log file will be generated. It's set to the bin folder by default and can be modified by the user to change it to a customized one.	The Bin folder path

#### **Table 3: Configuration File Properties**

However, it's much easier and safer to change the "ReportsFolder" settings in the Save Alarms Reports tab, instead of editing the .ini file yourself. We strongly suggest that you do not change the values in these optional sections of the .ini file, but instead, make any needed changes using the interface.

If you would like to automatically archive log files, select the Archive last log check box.



2099.19				
Trace Level	Detailed debugging informatic 💌			
	,			
And Andrive Lent Lee				
Archive Last Log				

Figure 32: Log Configuration Dialog

Logging		
Trace Level	Detailed debugging informatic 💌	
Archive La	Trace only errors Some extra information	
	Debugging information	
	Detailed debugging information	

Figure 33: Choosing the Log Level

# 2. Runtime Configuration

When running the application deploying the Integration Objects' Alarms Logger, if you right click on the control, the following menu appears (figure 30):

	View >
	Disconnect
	Show Achnowledge Dialog
•	Acknowledge
	Server Status
	Subscription Status
	Attributes >
	Available OPC Filters
	OPC Filters
	Enable/Disable Condition
	Filter Constraints
	Clear View
	Save
	Contact us
	About

Figure 34: Context Menu



The first popup menu appears when the application client is not connected to an OPC AE Server.

If you want to pick up the screen fields you want to view in the control layout, just click the *View...* menu item (Figure 31).





## 2.1. Connect to an OPC AE Server

To connect to an AE Server at runtime, choose the menu item "Connect..." from the popup menu. The following window will be displayed.

Browse your network to select the computer node on which the server is located and the ProgID AE server, and then click *Connect*.



Connect
Choose the OPC AE Server OPC AE Server : IntegrationObjects.OPC.AE.Knet.Server Server Address : DEV_04 Cancel Network Neighborhood Microsoft Windows Network Web Client Network DEV_04 DEV_04 IntegrationObjects.OPC.AE.Knet.Server IntegrationObjects.OPCAE.Simulation Matrikon.OPC.Simulation.1 Softing.OPCT oolboxDemo_ServerAE.1

Figure 36: "Connect" from Context Menu

## 2.2. Disconnect from an AE OPC Server

If your application is connected to an OPC AE Server and you want to disconnect from it, click *Disconnect* from the popup menu.

## 2.3. Acknowledge Alarms

This section provides information on acknowledging alarms:

To acknowledge an alarm requiring a response, the user may double click on the entry, or click the *Acknowledge* menu item from the context menu.

When the user right clicks on an alarm and chooses Acknowledge, the following dialog box will appear.



Alarm Ack			
Acknowledgment Action Acknowledges alarm currently Global Global Visible	OK Cancel		
Actor Identity Integration Objects' OPC Alarms Logger Comments :			
Acknowledged			
Don't show this dialog in the next time			

Figure 37: Acknowledging Alarms

In the Alarm Acknowledgment dialog box, users have the possibility to choose between three acknowledgment options which are listed in the table below.

Acknowledgment Options	Description
Point	Acknowledges alarms currently selected.
Global	Acknowledges all alarms received. This option quickly acknowledges all alarms from the current view.
Visible	Acknowledges all visible alarms. For example, if the size of the Viewer shows five alarms, and a total of eight alarms came in, only the five visible alarms are acknowledged.

#### Table 4: Acknowledgment Options

Users can also add comments and change the actor identity. If the user wants to always have the same acknowledgment options, he can click the *Don't show this dialog the next time* check box.

To display the acknowledgment options dialog once again, he may mark the "Show Acknowledge dialog" from the popup menu.



## 2.4. Server Information

To view the OPC server status, choose the "Server Status..." menu item in the popup menu (Figure 30) and the server status dialog box will appear (Figure 34). This window shows all static information about the AE Server.

The server status includes:

- Server start time
- Current time
- Time of last update sent
- Current OPC server state
- Major version of server
- Minor version of server
- Build number of the server
- Vendor specific information

Server Status	×	J
Server Connected State Server Time	IntegrationObjects.OPCAE.Simulation Yes, on \\DEV_04 Running. 25/09/2012 15:43:52 25/09/2012 15:44:01 me : 25/09/2012 15:44:00	
Version Major Version Minor Version Build Number	1     Integration Objects' OPC AE Simulation       0     Server	

Figure 38: Server Status

## 2.5. Event Subscription Status

To view the Event Subscription status, choose the "Subscription Status..." menu item in the context menu and the following dialog is displayed. This dialog allows users to get subscription state information and to modify the subscription state.



Event Subscription State			
Event Subscription State Buffer Time	✓ 1000		
Max Size	Close		

Figure 39: Event Subscription Status

The event subscription state is presented as a check box. When checked, the event subscription is active and it sends OnEvent notifications.

The buffer time is the time in milliseconds that the server can hold back the notification to buffer multiple events. Example, 1000 would indicate that the server buffer events along 1 second and sends the whole set of events as one callback.

The max size is the maximum number of events to buffer before sending events.

## 2.6. Select Returned Attributes

To retrieve the attributes of an existing Event Subscription, the user should select the Select Returned Attributes menu item. For each event category, SelectReturnedAttributes picks out the attributes to return. This method can be called many times in order to specify the attributes to return for each unique event type and event category pair. If this is called multiple times for the same event type and event category pair, then the latest call will be considered.

If you choose the "Select All Attributes for All Categories", all server event attributes will be displayed.

Attributes	►	Select All Attributes for All Categories
		Select Returned Attributes
		Get Returned Attributes



Selet Returned Attributes					
Event Type OPC	Event Type OPC_ALL_EVENTS				
Select an Event Ca	tegory				
Event Category	Description				
1	Level1				
2	Level2				
3	Level3				
4	Level4				
			- F		
Returned Attribute	s				
Attribute ID	Description	Attribute Type	•		
✓ 1	Attr 1	VT_I4	=		
2	Attr2	VT_I4			
<b>✓</b> 3	Attr3	VT I4			
	A 11. A	107 14	-		
	III	+			
4	Close				

Figure 40: Select Specific Server Returned Attributes

## 2.7. Retrieve Returned Attributes

To get the attributes, the user should click the *Get Returned Attributes* menu item. For each event category, the following dialog retrieves the attributes previously specified by the user in the Select Returned Attributes dialog.



Get Returned Attributes				
Event Type OPC_ALL_EVENTS				
Select an Event C	Category			
Event Category	Description			
1	Level1			
2	Level2			
3	Level3			
4	Level4			
	III	· ·		
Returned Attribut	tes			
Attribute ID	Description	Attribute Type 🔺		
1000	Attr 1000	VT_I4 =		
2000	Attr2000	VT_I4		
3000	Attr3000	VT_I4		
4000	A 11- 4000	107 14		
Apply				

Figure 41: Get Specific Server Returned Attributes

## 2.8. Enable/Disable condition by areas and sources



#### Figure 42: Enable/Disable Condition Menu Item

Enable condition by area:	Enable condition alarm events in an area.
Enable condition by source:	Enable condition alarm events for a given source.
Disable condition by area:	Disable condition alarm events in an area.
Disable condition by source:	Disable condition alarm events for a given source.

The dialog box below is the "enable condition by source." The user should navigate into the tree and check the wanted sources names in order to enable conditions for a given source.

The similar thing should be done if the user wants to enable a condition by area or to disable a condition by source or by area.



Enable Condition By Source
System_Event System_Event Tracking_EVENT Boiler 1 Water 1 Water 1
Apply Close

Figure 43: Enable Condition by Source

Enable Condition By Area			
Boiler 1 Water 1			
Apply Close			

Figure 44: Enable Condition by Area



Disable Condition By Source	X
System_Event Tracking_EVENT Boiler1 Boiler1:makeup1 Boiler1:makeup1 Boiler1:makeup2 Water1	
Apply Close	

Figure 45: Enable Condition by Area

Disable Condition By Area	X
Boiler1 Boiler1:makeup1 Boiler1:makeup2 Water1 Water1:makeup3 Water1:makeup4	_
Apply Close	

Figure 46: Enable Condition by Area

## 2.9. Available Filters

To view the filters of an OPC alarms and events server, the user should select the *Available OPC Filters* menu item and the following dialog screen appears:



Available OPC Filters			x	
Filter Mask	Available	Description	*	
OPC_FILTER_BY_SOURCE	Yes	The server supports filtering by source.		
OPC_FILTER_BY_AREA	Yes	The server supports filtering by area.	Ξ.	
OPC_FILTER_BY_SEVERITY	Yes	The server supports filtering by severity.		
OPC_FILTER_BY_CATEGORY	Yes	The server supports filtering by event categor		
OPC_FILTER_BY_EVENT	Yes	The server supports filtering by event type.	Ŧ	
•				
ОК				

Figure 47: Available Filters

## 2.10. OPC Filtering Configuration

To define OPC Filters, click the *OPC Filters* menu item from the context menu. The OPC Filtering dialog is used to set criteria for including or excluding specific alarms in the control. Users can specify the event types, severity, event category, areas or sources to be filtered.

The user can set filters on any event subscription in order to limit the events that he will be notified of. To setup a filter for an event subscription, the user can use the following criteria:

- **Filtering by event type**: only events satisfying the criterion "Event Type" will be returned.
- Filtering by event categories: only events satisfying the criterion "Event Categories" will be returned.
- **Filtering by areas and sources**: only events satisfying the criterion "Existing in these areas or having these sources" will be returned.
- **Filtering by Severity**: only events satisfying the criterion "Events that have a severity between the min and the max severity" will be returned.

Users can select multiple criteria; they will be logically ANDed together. This will cause all events satisfying these selected criteria to be returned.

If the user wants to get only specific event alarms, he may select the wanted check boxes (Figure 44).

The list of the event categories on the left indicates the categories to be filtered. The events on the right are the available categories. Select an event category and click on buttons (>> and <<) to move categories from one area to the other.



OPC Filtering	×	
Event Type     All Events     Condition     Simple     Tracking	Severity Low Severity 1 High Severity 1000	
Event Categories		
Areas	Sources SOURCE	
Add Remove	Add Remove	

Figure 48: Configure OPC Filtering Alarms

The list of areas indicates the areas containing the source names needing to be filtered. To add a new area or a new source, click the *Add* button and a dialog box appears (Figure 43 and Figure 46). The user may remove an area or a source by clicking the *Remove* button.



OPC Filtering				
Event	Type Browse OPC A	E Server		
Ever	⊙ Area O Source	Boiler 1:makeup2	-	
Area	Boiler 1 Boiler Boiler Boiler	r1:makeup1 r1:makeup2	-	
		OK Cancel		

Figure 49: Add Areas

OPC Filt	ering		
Eve	Browse OPC /	AE Server	-
	O Area	Boiler 1:makeup 1	
	⊙ Source	FIC1001	-
	abc System abc Tracking	_Event EVENT	
	Boiler 1		
	Are FIC1001		
Bo Water 1			
	1	OK Cancel	1
OK Cancel			

Figure 50: Add Sources



# 2.11. Apply Filter Constraints

By default, the AE Logger is configured to display all incoming alarms. To customize which alarms are displayed, right-click on the control, select *Filter Constraints...* and build an equation to restrict only the alarms that respond to the build filter constraint.

F	Filter Constraints					
	Enable Filter Constraints Constraints Set/Remove check mark to enable/disable filter					
	B	oolean Operator	Column Name	Operator	Value	
		AND	Event Time	=	25/09/2012 15	
		AND	Message	=	*Between?1	
		AND	Message	=	*Between?1	
	Add Constraint     Enable Constraint     Disable Constraint     Enable All       Disable All     Remove     Apply					

Figure 51: Filter Constraints Window at Runtime

## 2.12. About Box Dialog

Click *About* from the menu to display the About Box for the control. The About Box contains the product name and version number as well as other information about the software and Integration Objects.



#### OPC AE ActiveX User Manual



www.integrationobjects.com | customerservice@integrationobjects.com

#### Figure 52: About Box

# **3. Control Properties and Methods**

To edit the control properties, the user can simply use the control properties wizard and the runtime context menu without having to write any line code.

Property ServerName: Description: The OPC AE Server Name you want to connect to. Syntax: Property Public ServerName As String Example: AlarmsLogger .ServerName = "IntegrationObjects.OPCAE.Simulator" Property ServerAddress Description: The OPC AE Server location. Syntax: Property Public ServerAddress As String Example: AlarmsLogger .ServerAddress = "\\MyServer" Method Connect Description: Connect to the defined Server. Syntax : Public Function Connect() As Long

Example:

ServerName = "IntegrationObjects.OPCAE.Simulator" ServerAddress = "<u>\\127.0.0.1</u>" AlarmsLogger.Connect()

Method Disconnect



**Description**: Disconnect from the OPC Server. **Syntax** : Public Function Disconnect() As Long Example:

AlarmsLogger.Disconnect()

# 4. Control Events

## 4.1. OnEvent event

This event is posted to the application deploying the ActiveX whenever the control receives a message from the connected AE server. The only Argument within the event defines the event object. The event has the following syntax.

## 4.2. Public Event OnEvent(OpcEvent as Object)

To allow the application to obtain data from an event message posted by the connected AE Server, the user can simply double click the ActiveX control (VB environment) to view the event announcement. You may then add instructions to read the values for specific data contained within the event object.

Below is the detailed description of each data value for the event object.

## 4.2.1. Public Property AckRequired As Boolean

**Description**: True if the event requires acknowledgment.

#### 4.2.2. Public Property ActiveTime As Date

**Description**: The time that the event was active.

#### 4.2.3. Public Property EventTime As Date

Description: The time that the event was generated.

#### 4.2.4. Public Property ConditionName As String

**Description:** The condition name that caused the event to occur.

#### 4.2.5. Public Property SubConditionName As String

Description: The detailed cause of the event.

#### 4.2.6. Public Property EventCategory As Long

**Description**: One of the OPC event categories.

4.2.7. Public Property Cookie As Long

Description: The unique cookie associated with the event.

#### 4.2.8. Public Property Message As String

**Description**: The alarm message.



## 4.2.9. Public Property NewState As Long

**Description**: The type of alarm (active, acknowledged, ...). This property is based on these values:

```
IO_OPCAE_ConditionStateConstants
```

```
{
IO OPCAE CONDITION ENABLED
```

IO\_OPCAE\_CONDITION\_ACTIVE

IO\_OPCAE\_CONDITION\_ACKED

IO\_OPCAE\_CONDITION\_/

}

## 4.2.10. Public Property Quality As Long

**Description**: An indicator for the reliability of the event. This property is based on the following enumeration:

```
IO_OPCAE_QualityConstants
{
IO_OPCAE_QUALITY_BAD
IO_OPCAE_QUALITY_UNCERTAIN
IO_OPCAE_QUALITY_GOOD
}
```

Example:

Private Sub IOAlarmsLogger1\_OnEvent(OpcEvent As Object)

..

If (OpcEvent.Quality = IO\_OPCAE\_QUALITY\_BAD) Then MsgBox "Bad Quality"

End If

End Sub

## 4.2.11. Public Property StrQuality As String

**Description**: An indicator for the reliability of the event in a string format. This value is based on these string values: ("BAD", "UNCERTAIN", "UNKNOWN OPC QUALITY", "GOOD", "NON\_SPECIFIC")



Example:

Private Sub IOAlarmsLogger1\_OnEvent(OpcEvent As Object)

If (OpcEvent.StrQuality = "BAD") Then

MsgBox "Bad Quality"

End If End Sub

## 4.2.12. Public Property Severity As Long

**Description**: The severity of an alarm.

## 4.2.13. Public Property Source As String

Description: The source item that caused the event to occur.

## 4.2.14. Public Property EventType As Long

**Description**: OPC specific event type (simple, tracking, condition). This field is based on this enumeration:

IO\_OPCAE\_EventTypesConstants

{

IO\_OPCAE\_SIMPLE\_EVENT

IO\_OPCAE\_TRACKING\_EVENT

IO\_OPCAE\_CONDITION\_EVENT

IO\_OPCAE\_ALL\_EVENTS

}

Example:

Private Sub IOAlarmsLogger1\_OnEvent(OpcEvent As Object)

If (OpcEvent. EventType = IO\_OPCAE\_CONDITION\_EVENT) Then

End If End Sub

## 4.2.15. Public Property ActorID As String

**Description**: The name of the user that acknowledged the event.

#### 4.2.16. Public Property ChangeMask As Long

**Description**: The bits changed in the current event. The changed bits are below:



IO\_OPCAE\_ChangeConstants

{ IO\_OPCAE\_CHANGE\_ACTIVE\_STATE IO\_OPCAE\_CHANGE\_ACK\_STATE IO\_OPCAE\_CHANGE\_ENABLE\_STATE IO\_OPCAE\_CHANGE\_QUALITY IO\_OPCAE\_CHANGE\_SEVERITY IO\_OPCAE\_CHANGE\_SUBCONDITION IO\_OPCAE\_CHANGE\_MESSAGE IO\_OPCAE\_CHANGE\_ATTRIBUTE }

#### 4.2.17. Public Property EventAttributesCount As Long

**Description**: The number of attributes.

4.2.18. Public Property EventAttributes(Index As Integer) As Variant

Description: Vendor specific server attributes.

Example:

Private Sub IOAlarmsLogger1\_OnEvent(OpcEvent As Object) If (OpcEvent. EventAttributesCount <> 0) Then Dim I as Integer For I=0 to EventAttributesCount-1 SelectedAttributesList.AddItem CStr(EventAttributes(I))

Next

End If End Sub



# **CONFIGURING THE .NET AE LOGGER**

# **1. Runtime Configuration**

When running the application deploying the Integration Objects' OPC AE Alarm Logger .Net ActiveX, if you right click on the control, the following menu appears (figure 53):

	View	×
	Auto Load Configuration	
	Load Configuration	
	Clean Configuration	
	Connect	
	Disconnect	
~	Show Acknowledge Dialog	
	Acknowledge	
	Server Status	
	Subscription Status	
	Attributes	►
	Available OPC Filters	
	OPC Filters	
	Enable/Disable Condition	F
	Filter Constraints	
	Clear View	
	Print	•
	Save	
	Contact Us	
	About	
	Properties	

Figure 53: Context Menu

The first popup menu appears when the application client is not connected to an OPC AE Server.



## 4.3. View

If you want to pick up the screen fields you want to view in the control layout, just click the **View** menu item as shown below:

	View 🕨	~	Event Time
~	Auto Load Configuration	~	Active Time
	Load Configuration	~	Source
	Clean Configuration	~	Condition
	Connect	~	Sub Condition
	Disconnect	~	Message
~	Show Acknowledge Result	~	NewState
~	Show Acknowledge Dialog	~	Serverity
	Acknowledge	~	Quality
	Server Status	~	Ack Required
	Subscription Status	~	Actor ID
	Subscription status	~	Cookie
	Attributes	~	Event Type
	Available OPC Filters	~	Event Category
	OPC Filters	~	Ack Time
	Enable/Disable Condition	~	Ack Comments
	Filter Constraints		Attributes
	Clear View		
	Print		
	Save		
	Contact Us		
	About		
	Properties		

Figure 54: View Menu Item

# 4.4. Auto Load Configuration

In order to load a configuration automatically in the application startup you need to check the **Auto Load Configuration** item from the popup menu then restart the application.



Figure 55: Auto Load Configuration Menu Item



# 4.5. Load Configuration

In order to load the latest running configuration you should click on the **Load Configuration** item from the popup menu.

	View	•
	Auto Load Configuration	
	Load Configuration	
	Clean Configuration	
	Connect	
	Disconnect	
~	Show Acknowledge Result	
$\sim$	Show Acknowledge Dialog	
	Acknowledge	
	Server Status	
	Subscription Status	
	Attributes	F
	Available OPC Filters	
	OPC Filters	
	Enable/Disable Condition	►
	Filter Constraints	
	Clear View	
	Print	
	Save	
	Contact Us	
	About	
	Properties	

Figure 56: Load Configuration Menu Item

## 4.1. Clean Configuration

You can clean the configuration by clicking on the **Clean Configuration** item from the popup menu.



Figure 57: Clean Configuration Menu Item



## 4.2. Connect to an AE OPC Server

To connect to an AE Server choose the menu item **Connect** from the popup menu. The following window will be displayed.

	X	
Server Node	localhost	
Server Name IntegrationObjects.OPCAE.Simulation		
Get Servers Lis	Connect Cancel	

Figure 58: Connect From Context Menu

In order to connect to an OPC AE Server, you need to:

- Enter the server node of the OPC AE Server,
- Click on **Get Servers List** button to get all the available OPC AE Servers in the selected node, Select the ProgID AE Server from the returned server list then
- Click on the **Connect** button.

## 4.3. Disconnect from an AE OPC Server

If your application is connected to an OPC AE Server and you want to disconnect from it, click **Disconnect** from the popup menu.

Co	onnect
Di	sconnect

#### Figure 59: Disconnect Menu Item

## 4.4. Acknowledging alarms

To acknowledge an alarm requiring a response, the user may double click on the entry, or click the **Acknowledge** menu item from the context menu.

~	Show Acknowledge Result
4	Show Acknowledge Dialog
	Acknowledge

Figure 60: Acknowledge Menu Item

When the user right clicks on an alarm that requires acknowledgment and chooses Acknowledge, the following dialog box will appear.



📀 Alarm Ack		
Acknowledgment Action		
Acknowledges alarm currently selected.	Selected	
	Global	
	⊘ Visible	
Actor Identity Integration Objects' OPC Alarms Logger DotNet ActiveX Comment		
Acknowledged		
Do not show this dialog in the next time		
ОК	Cancel	

Figure 61: Acknowledging Alarms

In the Alarm Acknowledgment dialog box, users have the possibility to choose between three acknowledgment options which are listed below:

- Selected: Acknowledges alarm currently selected.
- **Global**: Acknowledges all alarms received. This option quickly acknowledges all alarms from the current view.
- Visible: Acknowledges all visible alarms. For example, if the size of the Viewer shows five alarms, and a total of eight alarms came in, only the five visible alarms are acknowledged

Users can also add comments and change the actor identity. If the user wants to always have the same acknowledgment options, he can click the *Do not show this dialog the next time* check box.

To display the acknowledgment options dialog once again, he may mark the **Show Acknowledge Dialog** from the popup menu.



#### Figure 62: Show Acknowledge Dialog Menu Item

To display the acknowledgment result message box in case the operation failed, he may mark the **Show Acknowledge Result** from the popup menu.





Figure 63: Show Acknowledge Result Menu Item

## 4.5. Server Information

To view the OPC server status, choose the **Server Status** menu item in the popup menu and the server status dialog box will appear. This window shows all static information about the AE Server.

The server status includes:

- Server start time
- Current time
- Time of last update sent
- Current OPC server state
- Major version of server
- Minor version of server
- Build number of the server
- Vendor specific information



₩ Server Sta	tus
Server	IntegrationObjects.OPCAE.Simulation
Connected	Yes,on \localhost
State	Running.
- Server Time -	
Start Time :	04/07/2016 14:12:24
Current Time	: 04/07/2016 14:12:40
Last Update T	ime : 04/07/2016 14:12:38
Version	
Major Version	1 2
Minor Version	0
Build Number	3
Vendor Inform	ation ojects' OPC AE Simulation Server

Figure 64: Server Status

## 4.6. Event Subscription Status

To view the Event Subscription status, choose the **Subscription Status** menu item in the context menu and the following dialog is displayed. This dialog allows users to get subscription state information and to modify the subscription state.

Event Subscription State		
V Event Sub	scription State	
Buffer Time	1000	-
Max Size	0	-
Apply Close		

Figure 65: Event Subscription State

The event subscription state is presented as a check box. When checked, the event subscription is active and it sends events notifications.



The buffer time is the time in milliseconds that the server can hold back the notification to buffer multiple events. Example, 1000 would indicate that the server buffer events along 1 second and sends the whole set of events as one callback.

The max size is the maximum number of events to buffer before sending events.

## 4.7. Select Returned Attributes

To retrieve the attributes of an existing Event Subscription, the user should select the **Select Returned Attributes** menu item. For each event category, Select Returned Attributes picks out the attributes to return. This method can be called many times in order to specify the attributes to return for each unique event type and event category pair. If this is called multiple times for the same event type and event category pair, then the latest call will be considered.

If you choose the **Select All Attributes for All Categories**, all server event attributes will be displayed.



#### Figure 66: Select All Attributes for All Categories Menu Item

Select Returned Attributes			
Event Type OPC_ALL_EVENTS			
Select an Event Category			
Event Category	Event Category Description		
1 time tick			
3	3 mouse click		
2	time slot		
Returned Attributes			
Attribute ID	Attribute ID Description Attribute Type		
✓ 1	current time	VT_DATE	
Apply Close			

Figure 67: Select Specific Server Returned Attributes



## 4.8. Retrieve Returned Attributes

To get the attributes, the user should click the **Get Returned Attributes** menu item. For each event category, the following dialog retrieves the attributes previously specified by the user in the Select Returned Attributes dialog.

🧭 Get Returned Attributes		
Event Type OPC_ALL_EVENTS		
Select an Event Category		
Event Category Description		
1	time tick	
3	mouse click	
2 time slot		
Returned Attributes		
Attribute ID	Description	Attribute Type
<b>V</b> 1	current time	VT_DATE
	Apply Close	

Figure 68: Get Specific Server Returned Attributes

# 4.9. Enable/Disable condition by areas and sources

Enable/Disable Condition	Enable Condition by Area
	Enable Condition by Source
	Disable Condition by Area
	Disable Condition by Source

#### Figure 69: Enable/Disable Condition Menu Item

- Enable condition by area: Enables condition alarm events in an area.
- Enable condition by source: Enables condition alarm events for a given source.
- Disable condition by area: Disables condition alarm events in an area.
- **Disable condition by source**: Disable condition alarm events for a given source.


The dialog box below is the "enable condition by source." The user should navigate into the tree and check the wanted sources names in order to enable conditions for a given source.

The similar thing should be done if the user wants to enable a condition by area or to disable a condition by source or by area.

Enable Condition By Source
e Boiler1
FIC1001
Boiler1:makeup2
FIC1002
ia
ia
FIC1003
⊡…  □ Water1:makeup4
FIC1004
Imperation Tracking_EVENT
OK Cancel

Figure 70: Enable Condition by Source

S Enable Condition By Area	J
Boiler1:makeup1 Boiler1:makeup2 Water1 Water1:makeup3 Water1:makeup4	
OK Cancel	

Figure 71: Enable Condition by Area



S Disable Condition By Area
Boiler1 Boiler1:makeup1 Boiler1:makeup2 Water1
Water1:makeup3
OK Cancel

Figure 72: Disable Condition by Area

O Disable Condition By Source
FIC1001
ia  Boiler1:makeup2
FIC1002
i i i i i i i i i i i i i i i i i i i
🛱 🖳 🔜 Water1:makeup3
FIC1003
Water1:makeup4
FIC1004
System_Event
Image Tracking_EVENT
OK Cancel

Figure 73: Disable Condition by Source

# 4.10. Available filters

To view the filters of an OPC alarms and events server, the user should select the **Available OPC Filters** menu item and the following dialog screen appears:



Y Available OPC Filters		×
Filter Mask	Available	Description
OPC_FILTER_BY_EVENT	Yes	The server supports filtering by event type.
OPC_FILTER_BY_CATEG	Yes	The server supports filtering by event category.
OPC_FILTER_BY_SEVER	Yes	The server supports filtering by severity.
OPC_FILTER_BY_AREA	Yes	The server supports filtering by area.
OPC_FILTER_BY_SOURCE	Yes	The server supports filtering by source.
		ОК

#### Figure 74: Available Filters

## 4.11. OPC Filtering Configuration

To define OPC Filters, click the **OPC Filters** menu item from the context menu. The OPC Filtering dialog is used to set criteria for including or excluding specific alarms in the control. Users can specify the event types, severity, event category, areas or sources to be filtered.

The user can set filters on any event subscription in order to limit the events that he will be notified of. To setup a filter for an event subscription, the user can use the following criteria:

- Filtering by event type: only events satisfying the criterion "Event Type" will be returned.
- Filtering by event categories: only events satisfying the criterion "Event Categories" will be returned.
- Filtering by areas and sources: only events satisfying the criterion "Existing in these areas or having these sources" will be returned.
- **Filtering by Severity**: only events satisfying the criterion "Events that have a severity between the min and the max severity" will be returned.

Users can select multiple criteria; they will be logically added together. This will cause all events satisfying these selected criteria to be returned.

If the user wants to get only specific event alarms, he may select the wanted check boxes.

The list of the event categories on the left indicates the categories to be filtered. The events on the right are the available categories. Select an event category and click on buttons (>> and <<) to move categories from one area to the other.



Y OPC Filtering	X
Event Type	Severity
All Events Condition	Low Severity 1
Simple Tracking	High Severity 1000
Event Categories	
Selected Event Categories	All Event Categories
level1	< Level1
	Level2
	> Level3
Areas	Sources
AREA	SOURCE
Add Remove	Add
ОК	Cancel

Figure 75: Configure OPC Filtering Alarms

The list of areas indicates the areas containing the source names needing to be filtered. To add a new area or a new source, click the **Add** button and a dialog box appears. The user may remove an area or a source by clicking the **Remove** button.



Y OPC Filtering	X
Event Type	Severity
All Events Condition	Low Severity 1
Simple Tracking	High Severity 1000
Event C Selecte	ver X
Area Boiler1:mal	keup1
Source	
⊡ Boiler1 Boiler1:mak Boiler1:mak ⊕ Water1	eup1 eup2
Areas	
AREA	
ОК	Cancel
Add Remove	Add Remove
ОК	Cancel

Figure 76: Add Areas



Y OPC F	iltering	2	8
Event Ty	pe	Severity	
V All Ev	ents Condition	Low Severity 1	•
Simple	e 🔄 Tracking	High Severity 1000	•
Event C Selecte	🧐 Browse OPC AE Sev	ver X	
	⊘ Area		
	Source FIC1001		
Areas AREA	Boiler1 Boiler1:make FIC1001 Boiler1:make Water1 System_Event Tracking_EVEN	eup1 l eup2 IT	
	ОК	Cancel	
Add	Remove	Add Remove	
	ОК	Cancel	

Figure 77: Add Sources

# 4.12. Apply Filter Constraints

By default, the AE Logger is configured to display all incoming alarms. To customize which alarms are displayed, right-click on the control, select **Filter Constraints** and build an equation to restrict only the alarms that respond to the build filter constraint.



Y Filter Constraints					
Cons	Constraints				
Set/	Remove check mark to e	enable/disable filter			
	Boolean Operator	Column Name	Operator	Value	
<b>V</b>	AND	EVENTTIME	=	04/07/2016 13:57:5	
1	AND	SEVERITY	>	100	
1	AND	QUALITY	=	GOOD	
< □					
(	Add Constraint Enable Constraint Disable Constraint Enable All				
Disable All Remove Apply					

Figure 78: Filter Constraints Window

### 4.13. Clear View

The end user may clear the list view by clicking on the **Clear View** item in the popup menu as shown in the below figure.

(	Clear View	
F	Print	F
5	Save	

Figure 79: Clear View Menu Item

# 4.14. Print

The end user can print the available alarms in the list view by clicking on the **Printer** item of the popup menu.

📄 Print Configuration			×
Page Setup	Print Preview	Print	✓ Fit to Page

#### Figure 80: Print Configuration

The Print Configuration dialog box shown above allows the following actions:

• Setup the page by clicking on the **Page Setup** button and choosing either the portrait or the landscape view



Page Setup	
	1 Contraction provide from the contraction of the c
Paper	
Size:	etter 🔹
Source:	utomatically Select
Orientation	Margins (millimeters)
Portrait	Left: 10 <u>Rig</u> ht: 10
C L <u>a</u> ndscape	<u>T</u> op: 10 <u>B</u> ottom: 10
	OK Cancel

Figure 81: Page Setup Dialog Box

• Preview the print before printing by clicking on the **Print Preview** button:





🚽 Prin	Print preview																
B 🔎	-	] 🔲		88		<u>C</u> lose									P	age	1
					- 11												-
																	_
						Int	egration	Objects'	OPC AE	E ActiveX							
	Paul Texa	Anton T.,	Raar on	Carillan	Rok Confilm	Manage	Namil Late	Rose by	Coully	Ash Rayle of	Anto 10	Cashie	Paul Spa	Post Calary	Auto Tana	Ash Cares	
2		- 202	anglada -	aria are	and a more	No cared	and the second	12	8	800		N CESH	CONCIDENCE.	Creation .			
-	20202-0		anglasis			The second se											
	33245		and the second s			The second se								1.4			
-			erende de			The same of		-									
-	-		and the second			-		-						1941			
-			energianale			No case of		-						1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -			
-	A	-	and a second			10000		-									
	33241	-	angla da			To cannot		e	-				1000	Crea Cath			
-	3327 C 1		angle de			To cannot							10.0				
0	33241		and shares and shares			The case of		•						Crea Cali			
-			any since			To see		-					107.0	-			
2	20241.		and the second s			The second as								100			
	S102161		erryde de			The same of		-									
-			-					-						199110			
			energia de			The same of								100 I I I			
-			and the second second					-					107.0	1991			
-	State of Street		and the second s			THE LOCAL		-									
-			energia de			The same of								100 I I I			
			and a second										107.0	199110			
31	33251		and the second s			The same of the		e					INPL I	Crea Call			
-	States and		and the second s			The second se											
27	3333 G C		erryde de			No cannol		e					INP.	0-a 66			
-	BB265.		any in the			To see .		-					117-11	1941			
e 8			and the second s			Personal		-						1.4			
	S10206 5		ereptende			The second											
-			and the second s					-									
4			energianeta			No cannot		e e						1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -			
	-					-		-									
4	33241		anglasis			Se		•					a me ca	0-0.00			
-			angle de			No cannot		-									
-	222262		anglasis			No case of		•						0-10A			
-	B2265		any instance			To see .		-					107.0	100			
-			and the second s			THE COLOR OF CALL		-									
-			errole de			The same of								Crea Cath			
-			any in the			To cannot		-						-			
Page 1																	

#### Figure 82: Preview Print Dialog Box

• Make the printing fits the page by checking the Fit to Page check box

Fit to Page

#### Figure 83: Fit to Page Check Box

• Choose the printer and print the alarms by clicking on the Print button



Pı	rint		
	Printer		
	<u>N</u> ame:	Microsoft XPS Document Writer	▼ <u>P</u> roperties
	Status:	Ready	
	Type:	Microsoft XPS Document Writer	
	Where: Comment:	XPSPort:	Drink to file
	comment.		
	Print range		Copies
	● <u>A</u> I		Number of <u>c</u> opies: 1
	Pages	<u>f</u> rom: <u>t</u> o:	
	Selection	n	
			OK Cancel

Figure 84: Choose Printer

## 4.15. Save

The end user can save the alarms in a csv file by clicking on the **Save** item of the popup menu.





# 4.16. Properties

To view or modify the .Net Alarm Logger properties, select the **Properties** item in the popup menu.



Figure 86: Properties Menu Item



#### 4.16.1. OPC Options Configuration

🔀 Properties				<b>×</b>
Alarms OPC Use this	Logger ( Property P	OPC Options age to define the OPC A	E Server Name and the AE	E Server Address.
Alarms Logger	Layout	Filter Constraints	Save Alarms Reports	ОК
OPC	Confi	gure Alarms Colors	Email Alarms	
OPC Connection	on h the OPC	AE Server		Cancel
Server Node	localh	ost		
Server Name	Integr	ationObjects.OPCAE.Sim	ulation	
			Get Servers List	
- Event Subscrip	tion			
Activate				
Buffer Time	1000	×		
Max Size	0	A V		
Alams Logger OPC OPC Connection Auto Launch Server Node Server Name Event Subscrip Activate Buffer Time Max Size	Layout Confi on h the OPC localh Integr tion	Filter Constraints gure Alarms Colors AE Server ost ationObjects.OPCAE.Sim	Save Alams Reports Email Alams	OK Cancel

#### Figure 87: Alarms Logger OPC Options

The above figure represents the first tab sheet of the properties dialog, which is used to define the OPC Connection and Event subscription parameters.

- OPC Connection: To pick up an AE server, you should enter the server node, click on the Get Servers List button then select the server ProgID from the returned OPC Server List.
- *Event Subscription*: To set the event subscription parameter, you can either activate or deactivate the alarm and enter the buffer time and max size values.

#### 4.16.2. OPC Alarm Color Configuration

Integration Objects' OPC AE Alarm Logger .Net ActiveX offers several options to configure GUI to be efficient and to respect the users' preferences. To change the configuration use the **Configure Alarms Colors** tab sheet in the main properties window.

The next tab sheet gives users the possibility to adjust background and text colors for alarms. Colors are an important piece of information for the alarm display.



You may also choose to enable blinking of active and unacknowledged alarms by clicking on the "enable blinking" check box.

× Properties						×		
Alarms OPC Click to	Logger adjust Bac	OPC Alar kround and	r <b>ms Colors</b> Text colors fo	r Alarms				
Alarms Logger	Layout	Filter Co	onstraints	Save	Alarms Reports	ОК		
OPC	Conf	igure Alarms	Colors		Email Alarms	Cancel		
Colors			Background	Color	Text Color	Cancer		
Active Alarm N	lot Acknow	ledged						
Active Alarm A	cknowledg	jed						
Inactive Alarm	Not Ackno	wledged				Color		×
Inactive Alarm	Acknowle	dged				Basic colors:		
Event Acknowl	edged							
Event Not Ackr	nowledged							
Set Old Ala	irms Color	S						
Blinking								
🔲 Blinking En	abled							
						Custom colors:		
						Define Cust	om Colors >>	
						OK Ca	incel	

#### Figure 88: Configure Alarms Colors

#### 4.16.3. Email Alarms Configuration

You can configure the application to send emails when an event or an alarm occurs. The user must provide the destination email address and SMTP server.

Integration Objects' OPC AE Alarm Logger .Net ActiveX has a tab sheet used to configure email notifications. When an alarm occurs, the application using the control automatically sends detailed notifications to the administrator by e-mail. The user can specify low or high severity for received alarms and events.



To enable mail notification, you should check the "enable mail notification" check box. If you choose to select the "send only critical alarms" option, you will receive only the incoming alarms that meet the severity level you define.

🔀 Propertie	s			×				
Alan Use alan	ms Logger ( this page to all m occurs.	OPC Alarms Notifie low the Alarms Logger to	cation Mails send emails to specific addr	esses when an				
Alarms Log	ger Layout	Filter Constraints	Save Alarms Reports	ОК				
OPC	Confi	gure Alarms Colors	Email Alarms					
Enable m Mail Server Smtp Server Sender mail Require User Name	OPC       Conligure Alarms Colors       Entail Addition         Imail Server and sender e-mail address       Smtp Server       smtpserver.com         Sender mail address       sales@integrationobjects.com       Imail Address         Imail Require Authentication       User Name       Imail Address							
Password								
	normation							
Name (	Lustomer Servic	be						
Email	Email customerservice@integrationobjects.com							
Send or	Send only critical alarms							
Low Severi	Low Severity 800							
High Sever	ity 1000		×					

#### Figure 89: Configure Email Notifications

#### 4.16.4. Control Layout Configuration

The Control layout tab is used to select the columns you want to appear on your control panel. The following dialog box allows users to view custom information, to pick up columns you want to display in the control panel, and to choose displayed format data to access further information.



A Properties	<b>X</b>
Alarms Logger Control Layout Use this page to modify the layout options. you are allowed to change colube viewed	mn information to
OPC Configure Alarms Colors Email Alarms	ок
Alarms Logger Layout Filter Constraints Save Alarms Reports	Cancel
Pick up the columns you want to see in your control	
Screen Fields	
Event Time ActorID Comments Condition	
Active Time Ack Time Cookie SubCondition	
Event Category Quality Ack Required Vew State	
Event Type 🔽 Source 🔽 Message 🗌 Attributes	
Severity	
Layout Options	
Allow column reordering Show grid lines Insert new alarms on top	
Data Format	
Event Time Date and Time  Ack Time Date and Time	
Active Time Date and Time Value Quality [Quality Value]	

Figure 90: Alarms Logger Layout

This ActiveX control includes the following columns:

Column Heading	Description		
Event Time	The time that the event was generated.		
Active Time	The time that the event was active.		
Source	The source item that caused the event to occur.		
Condition	The condition name that caused the event to occur.		
Sub Condition	The detailed cause of the event.		
Message	The alarm message.		
Event type	OPC specific event type (simple, tracking, condition).		
Event category	One of the OPC event categories.		
Quality	An indicator to the reliability of the event.		



Actor ID	The name of the user that acknowledged the event.		
Ack Comment	The comments entered by the user when he acknowledges the event.		
Ack required	Indicates if the event requires acknowledgment.		
Severity	The severity of an alarm.		
Ack Time	The time when the user acknowledged the alarm.		
Cookie			
NewState	The type of alarm (active, acknowledged).		
Attributes	Vendor specific server attributes.		

#### Table 5: The ActiveX Columns

If the user wants to pick up only some screen fields to display in the control, simply uncheck the unwanted columns.

All columns are checked by default except the "Attributes" column which the user may mark if so desired.

Pick up the columns you want to see in your control

Screen Fields			
V Event Time	ActorID	Comments	Condition
📝 Active Time	🔽 Ack Time	📝 Cookie	V SubCondition
Event Category	🔽 Quality	📝 Ack Required	📝 New State
V Event Type	Source	🔽 Message	Attributes
Severity			

Figure 91:Pick Up Columns to Display

This Layout dialog allows users to define layout options for the control.

The user can choose the ability to drag-and-drop column headers to reorder columns in the control. By default, drag-and-drop reordering of columns is enabled (Figure 94).



#### Figure 92: Enabling Reordering Columns

To move a column, click the column header once to select it. Then click and drag the column header to a new location (Figure 95).



		Event Time	Active Timetando	Condition	Source	Sub Condition	Message	NewState	*
6	;	04/07/2016 1	04/07/20	PVLEVEL	FIC1001	LO	Condition No	ENABLED	
7	, ,	04/07/2016 1	04/07/20	PVLEVEL	FIC1001	LOLO	Condition No	ENABLED	
8	}	04/07/2016 1	04/07/20	PVLEVEL	FIC1001	LOLO	Condition No	ENABLED	
Q		04/07/2016 1	04/07/20	PV/LEV/EL	EIC1001	1010	Condition No	ENABLED	

Figure 93: Drag-and-Drops Column Headers to Reorder Columns

We've decided to add a new preference on the layout page which allows you to select the time format for some screen field controls.

Along with choosing a custom date/time format, the user may choose to display date and time or just time for the three fields: **Event Time**, **Active Time** and **Ack Time**.

Format Data		
Event Time Date and Time -	Ack Time	Date and Time 🔹
Active Time Date and Time	Quality	[Description]

#### Figure 94: Customizing Data Format

#### 4.16.5. Filter Constraints Configuration

The Filter constraints tab is used to set criteria for including or excluding specific alarms in the list view.



🔀 Pro	perties			×					
*	Alarms Logger Filter Constraints Use this page to add make configurable filter constraints to view only certain types of alarms. You have the possibility to add constraints using wildcards.								
OF	PC Confi	gure Alarms Colors	Email Alarms	ОК					
Alam	ms Logger Layout	Filter Constraints	Save Alarms Reports						
Cons Set/	Constraints Set/Remove check mark to enable/disable filter								
	Boolean Operator	Column Name	Operator						
<b>V</b>	AND	EVENTTIME	=						
•			Þ						
Add	Add Constraint Enable Constraint Disable Constraint Enable All								
	Disable All Remove Apply								

Figure 95: Filter Constraints

To define a filter constraint, click the **Add Constraint** button and the dialog (Figure 98) will appear. You must select the filter column name, the operator (=, #, <, ...) and the value.

+ Add Constraint		×
New Constraint		
AND OR		
Column Name Operator	Value	
Event Time	04/07/2016 13:58:18	
Expression (EVENTTIME = 04/07/2016 13:58:18)		
OK Cance	el	

Figure 96: Add Constraint Dialog



Column Name
Event Time 🔹
Event Time
Active Time
Source
Condition
Sub Condition
Message
NewState
Severity
Quality
Ack Required
Cookie
Event Type
Event Category

Figure 97: Select the Constraint Field

Operator	
=	
=	
#	
<	
>	
<=	
>=	

Figure 98: Select the Constraint Operator

Before moving to another tab, you should click the *Apply* button to save modifications. If you did not do so, the following message box will appear.



Figure 99: Warning Message Box

#### 4.16.6. Save Alarm Reports Configuration

Integration Objects' OPC AE Alarm Logger .Net ActiveX allows the user to save the retrieved alarms into report files. To do that, the user has the possibility to define how frequently he wants the save to be performed (each minutes or after a certain number of minutes).

By default, alarms are saved each hour into the bin folder under the installation directory. If you are not pleased with this location, you can select another directory. Click **Browse Path** to select



the folder where you want to save the alarms. When saving the alarms report, the control creates a file in the chosen folder which is entitled:

"AlarmsReports{DAY]\_{MONTH]\_[YEAR]\_{HOUR]\_{MIN]\_{SECONDS]" with the "csv" extension.

× Properties		×	
Alarms Logger Save Alarms Report & Logging Use this page to configure the way you save alarms into files.			
OPC Co	nfigure Alarms Colors Email Alarms	ок	
Alarms Logger Layout	Filter Constraints Save Alarms Reports	Cancel	
Enable Archiver         Save alarms each       60         Browse file path         File Path       C:\Program Files (x86)\Integration Objects\Integra			
Logging			
Trace Level	Error  Auto append		
File Name	LogAELoggerActiveXDotNetControl		
Log File Path	C:\Program Files (x86)\Integration C Browse Path		
Buffer Size	200		
Auto Save TimeOut	10 ÷		
Maximum Files	5		

Figure 100: Save Alarms Reports



🔀 Properti	ies	23		
Ala	Alarms Logger Save Alarms Report & Logging			
U:	se this page to configure the way you save alarms into files.			
	Rechercher un dossier			
OPC	0	к		
Alarms L				
-Save Re		icel		
Save ala	A 📗 Integration Objects			
	Integration Objects' OPC AE ActiveX     Bin			
-Browse fi	Integration Objects' OPC AE ActiveX			
Eile Deth				
Flie Faul				
	🛛 🖉 Knet PI Bridge			
Logging	J mdbus			
Trace	🛛 🖉 MSOCache 🗸 🚽			
File Na				
Log Fi	Créer un <u>n</u> ouveau dossier OK Annuler			
Buffer	Size 200			
Auto Sa	ave TimeOut 10			
Maxim	um Files 0			

Figure 101: Select Reports Folder

## 4.17. Logging

Integration Objects' OPC AE Alarm Logger .Net ActiveX produces a log file named "LogAELoggerActiveXDotNetControl.LOG" that records errors and debugging information at runtime. If difficulties occur with the application, the log file can be extremely valuable for troubleshooting. When operations are running normally, the control will log very little information.

These log files are generated by default in the bin folder under the installation folder. These parameters all have default settings and can be changed by simply changing the information in the properties window.



Logging		
Trace Level	Error 👻	Auto append
File Name	LogAELoggerActiveXDotNet	Control
Log File Path	C:\Program Files (x86)\Integr	ation C Browse Path
Buffer Size	200	×
Auto Save TimeOut	10	×
Maximum Files	0	* *

#### Figure 102: Log File Setting

Log Setting	Description	Default Value
Buffer Size	The maximum number of messages to be stored in the runtime memory before launching writes action in the hard disk. It must be greater than 100.	200
Trace Level	The type of log messages to be logged. The value can be Control, Error, Warning, Inform, and Debug.	Error
AutoAppend	Set to true to continue writing log messages in the existed log file or to false to create a new file.	TRUE
Log File Path	The path where the log file will be generated. It's set to the bin folder by default and can be modified by the user to change it to a customized one.	The Bin folder path
Auto SaveTimeout	Time to wait to read all messages from the buffer	10
Maximum Files	Maximum number of files	5

#### Table 6: Log File Properties

## 4.18. About Box Dialog

Click **About** from the menu to display the About Box for the control. The About Box contains the product name and version number as well as other information about the software and Integration Objects.



About OPC AE Alarm Logger DotNet ActiveX		
integration objects	Integration OPC AE Ala DotNet Activ	Objects' Irm Logger /eX
© 2004 - 2018 Integration Objects		on Objects
	All rights reserved	
Web site: www.integrationobjects.com		Version 2.0.2
Sales: sales@integrationobjects.com		ок
Support: customerservice@integrationobjects.com		

Figure 103: About Box



# 2. Registering the OPC AE Logger .NET ActiveX

In order to register the Integration Objects' OPC AE Logger .Net ActiveX, the end user needs to follow the steps below:

- 1. Open the Command Prompt window as administrator.
- Locate the regasm.exe and IntegrationObjects.OPCAELoggerDotNetActiveX.dll already copied in your machine and copy the path
- 3. Type cd Path ( **Path** is the path of the regasm.exe and the IntegrationObjects.OPCAELoggerDotNetActiveX.dll
- 4. Type regasm IntegrationObjects.OPCAELoggerDotNetActiveX.dll /tlb:"C:\ IntegrationObjects.OPCAELoggerDotNetActiveX.tlb" /codebase



Figure 104: Registration Command Prompt

You can also use the **RegisterAELoggerDotNetActiveX** executable located in the bin folder under the installation directory. Once registered, you can add the **Activex.IOCSAELoggerActiveXCtrl** component in your workspace application.

# 3. Unregistering the OPC AE Logger .NET ActiveX

In order to unregister the Integration Objects' OPC AE Alarm Logger .Net ActiveX, the end user needs to follow the steps below:

- 5. Open the Command Prompt window as administrator.
- 6. Locate the regasm.exe and IntegrationObjects.OPCAELoggerDotNetActiveX.dll already copied in your machine and copy the path
- 7. Type cd Path ( **Path** is the path of the regasm.exe and the IntegrationObjects.OPCAELoggerDotNetActiveX.dll
- 8. Type regasm /u IntegrationObjects.OPCAELoggerDotNetActiveX.dll /tlb:"C:\ IntegrationObjects.OPCAELoggerDotNetActiveX.tlb" /codebase





Figure 105: Un-registration Command Prompt

You can also use the **UnregisterAELoggerDotNetActiveX** executable located in the bin folder under the installation directory.



# 4. Deploying the .Net AE Logger in Microsoft Visual Basic 6.0

# 4.1. Create a Standard EXE

Run the Microsoft Visual Basic 6.0 and select a Standard EXE to create a new project as shown below.



Figure 106: Create a VB6 Standard EXE

# 4.2. Add the OPC AE Net Logger reference

1. Select the Project menu item and click on References





Figure 107: Select Projet Reference

 Click on the Browse button and select the OPC AE Logger tlb file path (.:\Program Files (x86)\Integration Objects\Integration Objects' OPC AE ActiveX\Bin\DotNet\OPCAELoggerActiveX)



References - Project1	×
<u>A</u> vailable References:	ОК
✓ Visual Basic For Applications ✓ Visual Basic runtime objects and procedures	Cancel
Visual Basic objects and procedures	Browse
AccessControl 1.0 Type Library     AccessibilityCplAdmin 1.0 Type Library     Active DS Type Library	
ActiveMovie control type library ActiveX DLL to perform Migration of MS Repository V: AdHocReportingExcelClientLib	Help
AgControl 5.1 Type Library AP Client 1.0 HelpPane Type Library AP Client 1.0 Type Library	
۲	
OLE Automation	
Location: C:\Windows\system32\stdole2.tlb	
Language: Standard	

Figure 108: Browse the OPC AE Net Logger Path

3. Select the "IntegrationObjects.OPCAELoggerDotNetActiveX.tlb" and click on the *Open* button.

🚖, Add Reference	×		
Look in: 🚺 OPCAELoggerActiveX 🗸 🖛 🗈 📸 🎫			
Name	Date modified		
IntegrationObjects.Logger.SDK.dll	04/04/2016 09:36		
SintegrationObjects.OPCAELoggerDotNetActiveX.dll	30/01/2017 10:04		
IntegrationObjects.OPCAELoggerDotNetActiveX.tlb	30/01/2017 10:20		
IntegrationObjects.OPCNetClientSDK.dll	01/07/2016 12:47		
Size: 461	20/01/2017 15:16		
<	Þ		
File name: IntegrationObjects.OPCAELoggerDotNetActiveX	Open		
Files of type: Type Libraries (*.olb;*.tlb;*.dll)	Cancel		
	Help		

Figure 109: Select the type library (.tlb) file

4. Once the IntegrationObject\_OPCAELoggerDotNetActiveX reference is checked, click on OK



References - Project1	×	
<u>A</u> vailable References:	ОК	
<ul> <li>imapikor 1.0 Type Library</li> <li>IMContact 1.0 Type Library</li> <li>IMEAPI_JK 1.0 Type Library</li> <li>InfoPath Property Promotion Outlook UI 1.0 Type Lib</li> <li>InsRepIM 1.0 Type Library</li> <li>InstallShield Professional Setup Kernel D9.0</li> <li>InstallShield Script 1.0 Type Library</li> <li>InstallShield Script 1.0 Type Library</li> <li>InstallShield Setup Kernel</li> <li>InstallShield Windows Installer Setup Kernel 1.0 Type</li> <li>IntegrationObjects_OPCAELoggerDotNetActiveX</li> <li>IntegrationObjects_OPCEventClientDotNetActiveX</li> <li>IntegrationObjects_OPCEventClientDotNetActiveX</li> <li>IntegrationObjects_OPCEventClientDotNetActiveX</li> <li>Integlution iEIX ElectronicSignature 1.0 Type Library</li> </ul>	Cancel Browse	
IntegrationObjects_OPCAELoggerDotNetActiveX		
Location: C:\Program Files\Integration Objects\Integration Objects' OP( Language: Standard		
L		

#### Figure 110: Check the OPC AE Net Logger Reference

# 4.1. Add the OPC AE Net Logger Component to the Toolbar

5. Select the Project menu item and click on Components





Figure 111: Select Projet Components

6. Check the IntegrationObject\_OPCAELoggerDotNetActiveX component then click on the *OK* button



Components	×
Controls Designers Insertable Objects	
GigaSoft ProEssentials 3D Sci-Graph v6 GigaSoft ProEssentials Graph v6 GigaSoft ProEssentials Pie Chart v6 GigaSoft ProEssentials Polar-Smith v6 GigaSoft ProEssentials Sci-Graph v6 GridDTC HHActiveX 1.0 Type Library HybridAppControls iFIX Color Button Control module iFix Key Macro Editor Keyboard Control Integration Objects' OPC AE Controls Integration Objects' OPC AE Controls Integration Objects' OPC AE Controls Integration Objects' OPC AE Controls Integration Objects' OPC AELoggerDotNetActiveX	Browse
Location: C:\Windows\system32\mscoree.dll	
ОК	Cancel Apply

Figure 112: Select the OPC AE Net Logger Component

7. Select the IOCSAELoggerActiveXCtrl already added in the toolbar and draw the control in the form as shown below





#### Figure 113: Add the OPC AE Logger Component to the Form



# **USING OPC EVENT CLIENT ACTIVEX**

This control is a custom control which allows Visual Basic and other OLE Container applications to quickly and easily access data from any OPC Alarms & Events Server. The control's features enable the controlling application to establish a DCOM connection to an OPC A&E Server located on a local or on a remote machine. The application can subscribe to messages from the server and can be used to configure filters to receive only those of interest.

In this chapter, we will present the properties and methods of all objects in this Control.

# 1. IOEventClientCtrl

# 1.1. Logging

Integration Objects' Event Client produces a log file named "AELogEvent1.LOG" that records errors and debugging information. If difficulties occur with the application, the log file can be extremely valuable for troubleshooting. When operations are running normally, the control will log very little information.

This log file is generated by default in the bin folder under the installation folder. This path can be modified by the user using the "EventCltConfiguration.ini" configuration file incorporated by the toolkit which includes several logging parameters. These parameters all have default settings and can be changed at start-up by editing the configuration file, or by simply changing the information in the control properties window.



To change this file:

- 1. Open EventCltConfiguration.ini in a text editor.
- 2. Edit any of the parameters listed in the following tables:

Log Setting	Description	Default Value
LogFileMaxSize	The maximum log file size, in bytes. Once this size is reached during run-time, the log file is overwritten.	1048576*2 ~ 2 Mo (MegaByte)
TraceLevel	<ul> <li>The trace level is a value telling what type of information to log:</li> <li>0: Only errors messages are logged.</li> <li>1: Some extra information.</li> <li>2: Debugging information is logged.</li> <li>3: Detailed debugging information.</li> <li>The higher the trace level, the more information is recorded. We recommend you to use level 0 for a better performance of the client application.</li> </ul>	0
ArchiveLastLog	TRUE: Old file is copied to an intermediate file with incremental extension, before being overwritten. FALSE: Any pre-existing log file is erased and overwritten at start-up.	FALSE
LogFilePath	The path where the log file will be generated. It is set to the bin folder by default and can be modified by the user to change it to a customized one.	The Bin folder path

#### **Table 7: Configuration File Properties**

# 1.2. GetLocalOPCEventServers

GetLocalOPCEventServers		
Retrieve the list of local AE servers.		
listsrv = GetLocalOPCEventServers		
Return Value	Description	
listsrv	An array containing the progIDs (names) of registered local OPC AE Servers.	
Example		



Dim ServersList As Variant ServersList = GetLocalOPCEventServers ()



# 1.3. GetOPCEventServers

GetLocalOPCEventServers				
Retrieve the list of local AE Servers or AE Servers located on a remote machine.				
listsrv = GetOPCEventServers(host)				
Argument	Description			
host	The machine where the OPC Server is registered. Note that the host may be the IP address of the machine.			
Return Value	Description			
listsrv	An array containing the progIDs (names) of registered local OPC AE Servers.			
Example				
Dim ServersList As Variant ServersList = GetOPCEventServers ("localhost")				
Or				
ServersList = GetOPCEventServers (" <u>\\IP_Address</u> ")				
Or				
Dim HostName as String				
HostName = "\\IOServer"				
ServersList = GetOPCEventServers (HostName)				

# 1.4. CreateServer

CreateServer		
Add a new Server object to the IOEventClientCtrl ActiveX control.		
NewServer = CreateServer()		
Return Value	Description	
NewServer	This method returns a pointer to the newly created IOEventServer object.	



#### Example

Dim Server As New IOEventClientLib.IOEventServer Set Server = IOEventClient1.CreateServer ()

# 1.5. Servers

Servers				
This method sends a reference to an IOEventServer object, given the server index.				
Server = Servers ( index )				
Argument	Description			
index	The server index.			
Return Value	Description			
Server	This method returns a pointer to the IOEventServer object given by the server index. If the server with the given index is not found, this method returns NULL. This method also returns NULL if the Evaluation period has expired.			
Example				
Dim Server As IOEventClientLib.IOEventServer				
Dim Index as \	/ariant			
Index = 1				
Set Server = IC	DEventClient1.Servers (Index)			

# 1.6. AboutBox

AboutBox		
Display the About Box for the control. The About Box contains the product name and version number as well as other information about the software and Integration Objects.		
AboutBox()		
Example		
Call IOEventClient1.AboutBox ()		


## 2. IOEventServer

Property	Description
ComputerNode	[Read/Write]The machine where the OPC Server is registered. Note that the host may be the IP address of the machine.
ProgID	[Read/Write]The OPC Server name (ProgID).
ServerStatus	[ReadOnly]The current OPC AE Server Status Structure. This Structure is an <b>IOEventServerStatus</b> object.
CanFilterByEvent	[ReadOnly]A Boolean that defines the filter by Event.
CanFilterBySeverity	[ReadOnly] A Boolean that defines the filter by Severity.
CanFilterByCategory	[ReadOnly] A Boolean that defines the filter by Category.
CanFilterByArea	[ReadOnly] A Boolean that defines the filter by Area.
CanFilterBySource	[ReadOnly] A Boolean that defines the filter by Source.
CategoriesCount	[ReadOnly]Returns the number of categories defined within the A&E Server once the <b>QueryEventCategories</b> has been called.
ConditionsNumber	[ReadOnly]Returns the number of conditions related to the event category once <b>QueryConditionNames</b> has been called.
SubConditionsCount	[ReadOnly]Returns the number of subconditions related to the condition name once <b>QuerySubConditionNames</b> has been called.
SourceConditionsCount	[ReadOnly]Returns the number of conditions related to the concerned source name once <b>QuerySourceConditionNames</b> has been called.

### 2.1. ConnectToServer

#### ConnectToServer



Connect to the AE Server defined by <i>ProgID</i> and <i>ComputerNode</i> properties.			
LONG ConnectToServer (void) ; Public Function ConnectToServer( ) As Long			
Return Value	Description		
hr	The operation result.		
	Example		
Server.ComputerNode = " <u>\\IOServer</u> " Server.ProgID = "IntegrationObjects.AE.Simulation" Dim hr As Long hr = Server.ConnectToServer() If (hr = 0) Then MsgBox ("Connected to IntegrationObjects.AE.Simulation Server")			

### 2.2. Disconnect

Disconnect		
Disconnect from an AE Server that is already connected.		
LONG Disconnect (void) ; Public Function Disconnect () As Long		
Return Value	Description	
hr	The operation result.	
Example		
Dim hr As Long hr = Server.Disconnect () If (hr = 0) Then MsgBox ("Disconnected from IntegrationObjects.AE.Simulation Server") End If		

## 2.3. CreateEventSubscription

#### CreateEventSubscription



Create an event subscription object.		
IDispatch* CreateEventSubscription(VARIANT_BOOL Active, LONG BufferTime, LONG* RevisedBufferTime);		
Public Function CreateEventSubscription(ByVal Active As Boolean, _ ByVal BufferTime As Long, _ RevisedBufferTime As Long) As Object		
Argument	Description	
Active	Initial state of the event subscription, if true, the subscription will start sending events immediately.	
BufferTime	Time in milliseconds that the server can hold back the notification to buffer multiple events (e.g. 1000 would indicate that the server can buffer events for one second and send the set of events as one network call back).	
RevisedBufferTime	The revised buffer time that the server will actually support.	
Return Value	Description	
EventSub	The created event subscription object.	
Example		
Dim ESub As New IOEventClientLib.IOEventSubscription Dim RevisedBufTime As Long Set ESub = MyServer.CreateEventSubscription(True, 1000,RevisedBufTime) If Not (ESub Is Nothing) Then		
End If		

## 2.4. QueryEventCategories

QueryEventCategories	
Retrieve the available event categories for a set of event types.	
LONG QueryEventCategories(IO_OPCAE_EVENTTYPES_CONSTANTS EventType);	
Public Function QueryEventCategories(ByVal EventType As Long) As Long	
Argument Description	



EventType	The event type taken from the IO_OPCAE_EVENTTYPES_CONSTANTS enumeration
Return Value	Description
CatgeoriesCou	The returned long value is the number of received event categories. If the method failed, it returns 0. Apart from the returned argument, the <b>CategoryID</b> and <b>CategoryDescription</b> Arrays will be filled respectively with category IDs and category descriptions.
	Example
Dim count As Long count = MyServer.QueryEventCategories(IO_OPCAE_ALL_EVENTS) If (count > 0) Then Dim i As Integer ListView1.Clear For i = 0 To count - 1 Dim Iltem As ListItem Set Iltem = ListView1.ListItems.Add(i + 1, , CStr (MyServer.CategoryID( i))) Iltem.SubItems(1) = MyServer.CategoryDescription (i) Next End If	
'The EventType could be also a set of event types as follows :	
Dim EventType as Long EventType = IO_OPCAE_CONDITION_EVENT And _ IO_OPCAE_SIMPLE_EVENT	
count = MyServer. <b>QueryEventCategories</b> (EventType)	

## 2.5. CategoryID

	CategoryID
Returned array of event category ID's.	
Public CategoryID (ByVal Index As Long) As Long	
Argument	Description
Index	The index of the requested category. The index should be an integer between 0 and the categories count returned by the



QueryEventCategories function.		
Return Value	Description	
ID	The returned CategoryID given by the index.	
Example		
MsgBox CStr ( MyServer. CategoryID ( 2 ) )		

## 2.6. CategoryDescription

CategoryDescription		
Returned array of event category Descriptions.		
Public CategoryDescription (ByVal Index As Long) As String		
Argument	Description	
Index	The index of the requested category. The index should be an integer between 0 and the categories count returned by the <b>QueryEventCategories</b> function.	
Return Value	Description	
Description	The returned Category Description given by the index.	
Example		
MsgBox CStr (MyServer.CategoryDescription (2))		

# 2.7. QueryEventAttributes

QueryEventAttributes		
Retrieves the list of available event attributes for a given event category.		
IDispatch* QueryEventAttributes (LONG EventCategory);		
Public Function QueryEventAttributes (ByVal EventCategory As Long) As Object		
Argument	Description	
EventCategory	The given event category ID.	
Return Value	Description	



Object	The event attributes object : <b>IOEventAttributes</b> (details in IOEventAttributes Section)		
	Example		
Dim eAttr As IC	DEventClientLib.IOEventAttributes		
Set eAttr = My	Server. <b>QueryEventAttributes</b> (EventCategory)		
Dim j As Integ	er		
If (eAttr.Attribu	If (eAttr.AttributesCount <= 0) Then		
eAttributes	eAttributesForm.Caption = "No Attributes."		
Else			
For j = 0 To eAttr.AttributesCount - 1			
Dim II	Dim Iltem As ListItem		
Set IItem = eAttributesForm.ListView1.ListItems.Add(j + 1, , _			
CStr (eAttr. <b>AttributeIDs</b> (j)))			
<pre>Iltem.SubItems(1) = eAttr.AttributeDescriptions (j)</pre>			
lltem.	SubItems(2) = eAttr. <b>VarTypeString</b> (j)		
Next			
End If			

## 2.8. AckCondition

AckCondition			
Acknowledge the alarm	Acknowledge the alarm condition.		
LONG AckCondition (BSTR AcknowledgerID, BSTR Comment, BSTR Source, BSTR ConditionName, DATE ActiveTime, LONG ActiveTimeMilliseconds, LONG Cookie);			
Public Function AckCondition (ByVal AcknowledgerID As String, ByVal Comment As String, ByVal Source As String, ByVal ConditionName As String, ByVal ActiveTime As Date, ByVal ActiveTimeMilliseconds As Long, ByVal Cookie As Long) As Long			
Argument	Description		
AcknowledgerID	Identifying the name of acknowledging party.		
Source	The fully qualified source name.		
Comment	The comment provided by the acknowledger.		
ConditionName	Condition name.		



ActiveTime		The time the condition went active.
ActiveTimeMilliseconds		The milliseconds of the active time.
Cookie		The cookie value of the received event.
Return Value		Description
hr	The oper	ration result.
Example		
 IngResult = MyServer.AckCondition(ID, Comment, Source, Condition, ActiveTime, ActiveTimeMilliseconds, Cookie) If (IngResult = 0) Then MsgBox "Acknowledged" End If		

## 2.9. EnableConditionByArea

EnableConditionByArea		
Enable condition alarm events in an area.		
LONG EnableConditionByArea(BSTR Area);		
Public Function EnableConditionByArea (ByVal Area As String) As Long		
Argument	Description	
Area	Enable alarm area.	
Return Value	Description	
hr	The operation result.	
Example		
Dim Area As String		
Area = ""		
LngResult = MyServer.EnableConditionByArea (Area)		

## 2.10. EnableConditionBySource

#### EnableConditionBySource

Enable condition alarm events for a given source.



LONG EnableConditionBySource(BSTR Source);		
Public Function EnableConditionBySource (ByVal Source As String) As Long		
Argument	Description	
Source	Enable alarm source.	
Return Value	Description	
hr	The operation result.	



Example

Dim Source As String Source = "...." LngResult = MyServer.EnableConditionBySource ( Source)

## 2.11. DisableConditionByArea

DisableConditionByArea		
Disable condition alarm events in an area.		
LONG DisableConditionByArea(BSTR Area);		
Public Function DisableConditionByArea (ByVal Area As String) As Long		
Argument	Description	
Area	Disable alarm area.	
Return Value	Description	
hr	The operation result.	
Example		
Dim Area As String		
Area = ""		
LngResult = MyServer.DisableConditionByArea (Area)		

## 2.12. DisableConditionBySource

DisableConditionBySource	
Disable condition alarm events for a given source.	
LONG DisableConditionBySource(BSTR Source); Public Function DisableConditionBySource (ByVal Source As String) As Long	
Argument	Description
Source	Disable alarm source.
Return Value	Description



hr	The operation result.
Example	
Dim Source As String	
Source = ""	
LngResult = MyServer.DisableConditionBySource ( Source)	

## 2.13. QueryConditionNames

QueryConditionNames		
Retrieve the list of condition names for a specific event category.		
LONG QueryConditionNames (LONG EventCategory);		
Public Function (	QueryConditionNames (EventCategory As Long) As Long	
Argument	Description	
EventCategory	The event category ID.	
Return Value	Description	
Count	The number of the received condition names. You can read the returned array of condition names by the <b>ConditionName</b> function.	
Example		
count = MyServer. <b>QueryConditionNames</b> (EventCategory)		
If (count $> 0$ ) TI	If (count > 0) Then	
Dim i As Integer		
LstConditions.Clear		
For i = 0 To count - 1		
IstConditions.AddItem MyServer.ConditionName (i)		
Next		
End If	End If	

## 2.14. ConditionName



Argument	Description	
Index	The index of the requested condition names. The index should be an integer between 0 and the conditions count returned by the <b>QueryConditionNames</b> function.	
Return Value	Description	
CondName	Returned Condition Name.	
Example		
MsgBox MyServer.ConditionName (2)		

# 2.15. QuerySubConditionNames

QuerySubConditionNames		
Retrieves the list of subcondition names for a specific condition name.		
LONG QuerySubConditionNames (BSTR CondName); Public Function QuerySubConditionNames (CondName As String) As Long		
Argument	Description	
CondName	The condition name.	
Return Value	Description	
Count	The number of the received subcondition names. You can read the returned array of subcondition names with the <b>SubConditionName</b> function.	



Example
count = MyServer.QuerySubConditionNames (CondName)
If (count > 0) Then
Dim i As Integer
LstSubConditions.Clear
For i = 0 To count - 1
lstSubConditions.AddItem MyServer.SubConditionName (i)
Next
End If

## 2.16. SubConditionName

SubConditionName		
A property defining the returned array of subcondition names.		
BSTR SubCondit	ionName (LONG Index);	
Public Property	SubConditionName (ByVal Index As Long) As String	
Argument	Description	
Index	The index of the requested subcondition name. The index should be an integer between 0 and the subcondition's count returned by the <b>QuerySubConditionNames</b> function.	
Return Value	Description	
SubCondName	Returned SubCondition Name.	
Example		
MsgBox MyServer. ConditionName ( "Between" )		



## 2.17. QuerySourceConditionNames

QuerySourceConditionNames		
Retrieve the list of condition names for a given source name.		
LONG QuerySou Public Function	urceConditionNames (BSTR Source); QuerySourceConditionNames (Source As String) As Long	
Argument	Description	
Source	The source name.	
Return Value	Description	
Count	The number of the received condition names. You can read the returned array of condition names with the <b>SourceConditionName</b> function.	
	Example	
count = MyServer.QuerySourceConditionNames(EventCategory)		
If (count > 0) Then		
Dim i As Integer		
LstConditions.Clear		
For i = 0 To count - 1		
IstConditions.AddItem MyServer.SourceConditionName (i)		
Next	Next	
End If		

## 2.18. SourceConditionName

	SourceConditionName	
A property defining the returned array of source condition names.		
BSTR SourceConditionName (LONG Index);		
Public Property	SourceConditionName (ByVal Index As Long) As String	
Argument	Description	
Index	The index of the requested source condition name. The index should be an integer between 0 and the subcondition's count returned by the <b>QuerySourceConditionNames</b> function.	
Return Valu	e Description	



SourceCondName Returned source condition name.

Example

MsgBox MyServer.SourceConditionName (3)

## 2.19. GetConditionState

GetConditionState		
Get condition state object for a given source's condition.		
IDispatch* GetConditionState(BSTR Source, BSTR ConditionName, LONG* AttributeIDs, LONG AttributeIDsCount);		
Public Function String, Attributel IOEventConditio	Get( Ds A nSta	ConditionState (ByVal Source As String, ByVal ConditionName As As Long, ByVal AttributeIDsCount As Long) As ate
Argument		Description
Source		Source name.
ConditionName	÷	Condition name.
AttributeIDs		Array of requested attributes IDs.
AttributeIDsCo	unt	Number of attributes IDs.
Return Value		Description
Object	An	IOEventConditionState object.



#### 2.20. CreateBrowser

CreateBrowser		
Create an event browser object for the server.		
IDispatch* Creat Public Function	teBrowser(void); CreateBrowser() As IOEventBrowser	
Return Value	Description	
Browser	This method returns the IOEventBrowser object related to the connected server.	
Example		
Dim Browser As New IOEventBrowser Set Browser = MyServerCreateBrowser		

# 3. IOEventSubscription

Using the following properties, the user can set up filtering options for event subscription and retrieve the events.

Property	Description
FilterEventType	(Read/Write) A Bit mask indicating the selected event types that should be sent to this subscription, built by combining values from the <b>IO_OPCAE_EVENTTYPES_CONSTANTS</b> enumeration. Only events satisfying the criterion "FilterEventType" will be returned.
FilterHighSeverity	(Read/Write) The Highest Severity limit for this subscription (Between 1 and 1000) and must be equal to or greater than FilterLowSeverity.
FilterLowSeverity	(Read/Write) The Low Severity limit for the subscription (Between 1 and 1000) and must be equal to or less than FilterHighSeverity.
FilterCategoriesCount	(Read-only) The number of category IDs. If 0, all categories are selected.



FilterSourcesCount	(Read-only) The number of sources. If 0, all sources are selected.
FilterAreasCount	(Read-only) The number of areas. If 0, all areas are selected.
FilterCategories	(Read-only) Array of Category ID's (see QueryEventCategories method) that should be sent to the event subscription object.
FilterAreas	(Read-only) List of area names.
FilterSources	(Read-only) List of source names.

## 3.1. GetSubscriptionState

	GetSubscriptionState
Get subscription state information.	
void GetSubscriptionState(VARIANT_BOOL* Active, LONG* BufferTime, LONG* MaxSize, LONG* hClientSubscription);	
Public Sub GetSubscr Long, hClientSubscrip	iptionState (Active As Boolean, BufferTime As Long, MaxSize As tion As Long)
Argument	Description
Active	When Active is true, the event subscription is active and it sends the OnEvent notifications.
BufferTime	Time in milliseconds that the server can hold back the notification to buffer multiple events.
MaxSize	The maximum number of events to buffer before sending events.
hClientSubscription	The client handles for the subscription.
	Example
Call MyEventSub.GetSubscriptionState (Active, BufferTime, MaxSize, hClientSubscription) MsgBox CStr (BufferTime)	

## 3.2. SetSubscriptionState

#### SetSubscriptionState

Modify the subscription state information.



void SetSubscriptionState(VARIANT\_BOOL Active, LONG BufferTime, LONG MaxSize, LONG hClientSubscription, LONG\* RevisedBufferTime, LONG\* RevisedMaxSize);

Public Sub SetSubscriptionState (ByVal Active As Boolean, ByVal BufferTime As Long, ByVal MaxSize As Long, ByVal hClientSubscription As Long, RevisedBufferTime as Long, RevisedMaxSize As Long)

Argument	Description	
Active	When Active is true, the event subscription is active and it sends the OnEvent notifications.	
BufferTime	Time in milliseconds that the server can hold back the notification to buffer multiple events.	
MaxSize	The maximum number of events to buffer before sending events.	
hClientSubscription	The client handles for the subscription.	
RevisedBufferTime	The revised buffer time that the server will support.	
RevisedMaxSize	The revised maximum size that the server will support.	
Example		
Call MyEventSub Se	Call MyEventSub SetSubscriptionState (True 1000 3000 1	

Call MyEventSub.SetSubscriptionState (True, 1000, 3000, 1, RevisedBufferTime, RevisedMaxSize) MsgBox CStr (RevisedBufferTime)

## 3.3. Refresh

	Refresh
Refresh the subscription event.	
void Refresh ( void ); Public Sub Refresh ( )	

## 3.4. CancelRefresh

CancelRefresh
Cancel the Refresh of the subscription event.
void CancelRefresh ( void );
Public Sub CancelRefresh ()



#### 3.5. Activate

Activate	
Activate the subscription event.	
void Activate ( void ); Public Sub Activate ( )	

#### 3.6. Deactivate

Deactivate
Deactivate the subscription event.
void Deactivate ( void ); Public Sub Deactivate ( )

## 3.7. SelectAllAttributesForAllCatgories

#### SelectAllAttributesForAllCatgories

Selects all event attributes for all categories to be returned.

void SelectAllAttributesForAllCatgories ( void );

Public Sub SelectAllAttributesForAllCatgories ()



#### 3.8. GetReturnedEventAttributes

	GetReturnedEventAttributes
Selects all event attributes for all categories to be returned.	
IDispatch* GetReturnedEventAttributes (LONG EventCategory); Public Function GetReturnedEventAttributes (ByVal EventCategory As Long ) As IOEventAttributes	
Argument	Description
EventCategory	The event Category ID for which the caller wishes to get the returned attributes.
Return Value	Description
IOEventAttributes Object	An IOEventAttributes object.

## 3.9. SelectReturnedEventAttributes

GetReturnedEventAttributes	
Selects event attributes for a specific category to be returned.	
void SelectReturnedEventAttributes (LONG EventCategory, LONG* AttributeIDs, LONG Count); Public Sub SelectReturnedEventAttributes (ByVal EventCategory As Long , AttributeIDs As Long, Count As Long)	
Argument	Description
EventCategory	The event Category ID.
AttributeIDs	The array of attribute ID's.
Count	The number of attributes.



#### 3.10. GetFilter

GetFilter
Get OPC Filter options for this event subscription.
void GetFilter ( void ); Public Sub GetFilter ( )
Example
Call MyEventSub.GetFilter
Text1.Text = Str (MyEventSub. <b>FilterLowSeverity</b> ) Text2.Text = Str (MyEventSub. <b>FilterHighSeverity</b> )
List1.Clear List2.Clear Dim i As Integer If (MyEventSub. <b>FilterSourcesCount</b> > 0) Then For i = 0 To MyEventSub. <b>FilterSourcesCount</b> - 1 List2.AddItem MyEventSub. <b>FilterSources (i)</b> Next i End If
If (MyEventSub. <b>FilterAreasCount</b> > 0) Then For i = 0 To MyEventSub. <b>FilterAreasCount</b> - 1 List1.AddItem MyEventSub. <b>FilterAreas (i)</b> Next i End If



## 3.11. ApplyFilter

ApplyFilter
Set up the filtering options for this event subscription.
void ApplyFilter ( void ); Public Sub ApplyFilter ( )
Example
MyEventSub.FilterHighSeverity = 1000 MyEventSub.FilterLowSeverity = 800 MyEventSub.FilterEventType = IO_OPCAE_CONDITION_EVENT MyEventSub.ClearFilterAreas
MyEventSub. <b>ClearFilterSources</b>
Dim i As Integer For i = 0 To List1.ListCount - 1 MyEventSub. <b>AddFilterArea</b> (List1.List(i)) Next
For i = 0 To List2.ListCount - 1 MyEventSub. <b>AddFilterSource</b> (List2.List(i)) Next
Call MyEventSub. <b>ApplyFilter</b>



#### 3.12. AddFilterSource

	AddFilterSource	
Add a source name to the array of filtering source names.		
void AddFilterSource (BSTR Source); Public Sub AddFilterSource (ByVal Source As String)		
Argument	Description	
Source	The source name added to the filter source list.	
Example		
Dim Source As String Source = "Tag.AE.1" AddFilterSource ( Sou	rce)	

## 3.13. AddFilterArea

	AddFilterArea	
Add an area name to t	he array of filtering area names.	
void AddFilterArea (BSTR Area); Public Sub AddFilterArea (ByVal Area As String )		
Argument	Description	
Area	The area name added to the filter areas list.	
Example		
Dim Area As String Area = "computer" AddFilterArea ( Area)		



## 3.14. AddFilterCategory

AddFilterCategory		
Add a category ID to the array of filtering categories IDs.		
void AddFilterCategory (LONG CategoryID); Public Sub AddFilterCategory (ByVal CategoryID As Long )		
Argument	Description	
CategoryID	The category ID	
Example		
Dim ID As Long		
ID = 346		
AddFilterCatgeory	AddFilterCatgeory (ID)	

## 3.15. RemoveFilterSource

RemoveFilterSource		
Remove a source name from the source list.		
void RemoveFilterSource (LONG Index); Public Sub RemoveFilterSource (ByVal Index As Long )		
Argument	Description	
Index	A given index that references the index of the source in the list to be removed. The index is between 0 and the source count $-1$ .	
Example		
RemoveFilterSource (1)		

### 3.16. RemoveFilterArea

RemoveFilterArea
Remove the area name from the area list.
void RemoveFilterArea (LONG Index); Public Sub RemoveFilterArea (ByVal Index As Long )



Argument	Description	
Index	A given index that references the index of the area in the list to be removed. The index is between 0 and the area count $-1$ .	
Example		
RemoveFilterArea (1)		

# 3.17. RemoveFilterCategory

RemoveFilterCategory		
Remove the category ID from the category list.		
void RemoveFilterCategory (LONG Index); Public Sub RemoveFilterCategory (ByVal Index As Long )		
Argument	Description	
Index	A given index that references the index of the category in the list to be removed. The index is between 0 and the category count $-1$ .	
Example		
RemoveFilterCate	geory (1)	



## 3.18. ClearFilterSources

ClearFilterSources	
Clear the source list.	
void ClearFilterSources ( void ); Public Sub ClearFilterSources ( )	

## 3.19. ClearFilterAreas

ClearFilterAreas
Clear the area list.
void ClearFilterAreas ( void );
Public Sub ClearFilterAreas ()

## 3.20. ClearFilterCategories

ClearFilterCategories
Clear the category list.
void ClearFilterCategories (void);
Public Sub ClearFilterCategories ()

# 4. IOEventConditionState

Property	Description
Acknowledged	(Read-only) True if the condition is acknowledged.
AcknowledgerID	(Read-only) The ID string for the last acknowledger of the condition.
Active	(Read-only) True if the condition is active.
ActiveSubCondition	(Read-only) The name of the active subcondition.



Comment	(Read-only) The comment string provided by the last acknowledger of the condition.
Quality	(Read-only) The OPC quality of the condition source.
LastAckTime	(Read-only) The time the condition was last acknowledged.
State	(Read-only) The condition state.
ActiveSubConditionDefinition	(Read-only) The active subcondition definition.
ActiveSubConditionDescriptio	(Read-only) The active subcondition description.
ActiveSubConditionSeverity	(Read-only) The active subcondition severity.
SubConditionLastActiveTime	(Read-only) The time that the subcondition last became active.
ConditionLastActiveTime	(Read-only) The time that the condition last became active.
ConditionLastInactiveTime	(Read-only) The time that the condition last became inactive.
StateString	(Read-only) A string representation of the condition state.
SubConditionsCount	(Read-only) Number of subconditions.
QualityString	(Read-only) A string representation of quality string.
SubConditionDefinitions	(Read-only) The definitions of the condition's subconditions.
SubConditionDescriptions	(Read-only) The descriptions of the condition's subconditions.
SubConditionNames	(Read-only) The names of the condition's subconditions.
SubConditionSeverities	(Read-only) The severities of the condition's subconditions.

# 5. IOEvent

Property	Description
AckRequired	(Read-only) A Boolean that defines whether the alarm condition requires Acknowledgment or not.



ActiveTime	(Read-only) The time the condition alarm went active.
EventTime	(Read-only) The time when the event occurred.
ConditionName	(Read-only) The condition name (condition events).
SubConditionName	(Read-only) The subcondition name (condition events).
EventCategory	(Read-only) Event category ID.
Cookie	
Message	(Read-only) The alarm message.
NewState	(Read-only) Bit mask that indicates the condition's current state, based on IO_OPCAE_CONDITIONSTATE_CONSTANTS (Enumerations section)
Quality	(Read-only) The quality value for condition events.
Severity	(Read-only) The severity value for condition events.
Source	(Read-only) The source item that caused the event to occur.
EventType	(Read-only) The event type. The allowable values are values from the enumeration <b>IO_OPCAE_EVENTTYPES_CONSTANTS</b> . (Enumerations section)
ActorID	(Read-only) The acknowledger name (condition events).
ChangeMask	(Read-only) Bit mask indicating which items have changed. This property is based on <b>IO_OPCAE_CHANGE_CONSTANTS</b> (Enumerations section)
StrQuality	(Read-only) String representation of condition event quality.

StrQuality	(Read-only) String representation of condition event quality.
EventTimeMilliseconds	(Read-only) The milliseconds in the event time.
ActiveTimeMilliseconds	(Read-only) The milliseconds in the active time.
EventAttributesCount	(Read-only) The number of event attributes.
EventAttributes	(Read-only) Array of event attributes' values.
EventAttributesAsString	(Read-only) Array of event attributes' values as string.



# 6. IOEventServerStatus

Property	Description
ServerState	(Read-only)The current status of the AE Server. Allowable values are given by the enumeration : IO_OPCAE_SERVERSTATE_CONSTANTS (see the following table).
CurrentTime	(Read-only)The current time of the OPC AE Server.
StartTime	(Read-only)The time when the server process started.
LastUpdateTime	(Read-only)The last updated time of the server process.
MajorVersion	(Read-only)The major version identification of the historian.
MinorVersion	(Read-only)The minor version identification of the historian.
BuildNumber	(Read-only)The build number identification of the historian
ServeStateString	(Read-only)A string explaining Server status instead of enumeration values.
VendorInfo	(Read-only)The vendor information for the OPC Server.
CurrentTimeMilliseconds	(Read-only)The milliseconds of the current time of the OPC AE Server.
StarttimeMilliseconds	(Read-only)The milliseconds of the time when the server process started.
LastUpdateTimeMilliseconds	(Read-only)The last updated time in milliseconds.

#### Example

Dim SStatus As New IOEventClientLib.IOEventServerStatus Set SStatus =.MyServer.ServerStatus If Not (SStatus Is Nothing) Then IstStatus.Clear

Dim T as String T = CStr (SStatus.CurrentTime) + "." + CStr (SStatus.CurrentTimeMilliseconds) IstStatus.AddItem "Current Time : " + T





# 7. IOEventBrowser

Property	Description
SelectedArea	(Read-only) The selected area in the browser.
SelectedSource	(Read-only) The selected source in the browser.



# 7.1. ShowBrowserDialog

ShowBrowserDialog	
This method allows users to access a browser dialog to browse all address space and to select the source or area you want.	
LONG ShowBrowserDialog (void) ; Public Function ShowBrowserDialog ( ) As Long	
Example	
LngResult = Browser.ShowBrowserDialog () If (LngResult = 0) Then MsgBox Browser.SelectedArea MsgBox Browser.SelectedSource	
End If	

## 8. IOEventAttributes

Property	Description
AttributesCount	(Read-only) The number of attributes in the structure.
AttributeIDs	(Read-only) Array of the attributes' IDs.
AttributeDescriptions	(Read-only) Array of the attributes' Descriptions.
AttributeVarTypes	(Read-only) Array of the attributes' Data Types.
VarTypeString	(Read-only) Array of the attributes' Data Types in a string format.

## 9. ActiveX Defined Enumerations

## 9.1. IO\_OPCAE\_SERVERSTATE\_CONSTANTS Enumeration

Values
IO_OPCAE_STATUS_RUNNING
IO_OPCAE_STATUS_FAILED
IO_OPCAE_STATUS_NOCONFIG
IO_OPCAE_STATUS_SUSPENDED
IO_OPCAE_STATUS_TEST



IO\_OPCAE\_STATUS\_COMM\_FAULT

### 9.2. IO\_OPCAE\_FILTER\_CONSTANTS Enumeration

	Values
IC	D_OPCAE_FILTER_BY_EVENT
IC	D_OPCAE_FILTER_BY_CATEGORY
IC	D_OPCAE_FILTER_BY_SEVERITY
IC	D_OPCAE_FILTER_BY_AREA
IC	D_OPCAE_FILTER_BY_SOURCE

#### 9.3. IO\_OPCAE\_EVENTTYPES\_CONSTANTS Enumeration

Values
IO_OPCAE_SIMPLE_EVENT
IO_OPCAE_TRACKING_EVENT
IO_OPCAE_CONDITION_EVENT
IO_OPCAE_ALL_EVENTS

#### 9.4. IO\_OPCAE\_CONDITIONSTATE\_CONSTANTS Enumeration

Values
IO_OPCAE_CONDITION_ENABLED
IO_OPCAE_CONDITION_ACTIVE
IO_OPCAE_CONDITION_ACKED

#### 9.5. IO\_OPCAE\_CHANGE\_CONSTANTS Enumeration

Values
IO_OPCAE_CHANGE_ACTIVE_STATE
IO_OPCAE_CHANGE_ACK_STATE
IO_OPCAE_CHANGE_ENABLE_STATE
IO_OPCAE_CHANGE_QUALITY
IO_OPCAE_CHANGE_SEVERITY
IO_OPCAE_CHANGE_SUBCONDITION
IO_OPCAE_CHANGE_MESSAGE
IO_OPCAE_CHANGE_ATTRIBUTE



# USING OPC EVENT CLIENT .NET ACTIVEX

This control is a custom control which allows Visual Basic and other OLE Container applications to quickly and easily access data from any OPC Alarms & Events Server. The control's features enable the controlling application to establish a DCOM connection to an OPC A&E Server located on a local or on a remote machine. The application can subscribe to messages from the server and can be used to configure filters to receive only those of interest.

In this chapter, we will present the properties and methods of all objects in this Control.

## 10. IOEventClientDotNetCtrl

#### 10.1. Logging

Integration Objects' Event Client .Net ActiveX produces a log file named "OPCEventClientDotNetActiveX.LOG" that records errors and debugging information. If difficulties occur with the application, the log file can be extremely valuable for troubleshooting. When operations are running normally, the control will log very little information.

This log file is generated by default in the bin folder under the installation folder. This path can be modified by the user using the "OPCEventClientDotNetActiveXConfig.ini" configuration file incorporated by the toolkit which includes several logging parameters. These parameters all have default settings and can be changed at start-up by editing the configuration file.



To change this file:

- 1. Open OPCEventClientDotNetActiveXConfig.ini in a text editor.
- 2. Edit any of the parameters listed in the following tables:

Log Setting	Description	Default Value
Buffer Size	The maximum number of messages to be stored in the runtime memory before launching writes action in the hard disk. It must be greater than 100.	200
Trace Level	The type of log messages to be logged. The value can be Control, Error, Warning, Inform, and Debug.	Error
AutoAppend	Set to true to continue writing log messages in the existed log file or to false to create a new file.	TRUE
Log File Path	The path where the log file will be generated. It's set to the bin folder by default and can be modified by the user to change it to a customized one.	The Bin folder path
Auto SaveTimeout	Time to wait to read all messages from the buffer	10
Maximum Files	Maximum number of files	5

#### **Table 8: Configuration File Properties**

#### 10.2. GetLocalOPCEventServers

GetLocalOPCEventServers		
Retrieve the list of local AE servers.		
listsrv = GetLocalOPCEventServers		
Return Value	Description	
listsrv	An array containing the progIDs (names) of registered local OPC AE Servers.	

#### **10.3. GetOPCEventServers**

GetLocalOPCEventServers			
Retrieve the list of local AE Servers or AE Servers located on a remote machine.			
listsrv = GetOP	listsrv = GetOPCEventServers(host)		
Argument	Description		
host	The machine where the OPC Server is registered. Note that the host may be the IP address of the machine.		
Return Value	Description		
listsrv	An array containing the progIDs (names) of registered local OPC AE Servers.		

#### 10.4. CreateServer

CreateServer			
Add a new Server object to the IOEventClientDotNetCtrl ActiveX control.			
NewServer = 0	NewServer = CreateServer()		
Return Value	Description		
NewServer	This method returns a pointer to the newly created OPCAEServer object.		



#### 10.5. GetServer

Servers		
This method sends a reference to an OPCAEServer object, given the server index.		
Server = GetSe	erver(index)	
Argument	Description	
index	The server index.	
Return Value	Description	
GetServer	This method returns a pointer to the OPCAEServer object given by the server index. If the server with the given index is not found, this method returns NULL. This method also returns NULL if the Evaluation period has expired.	

#### 10.6. AboutBox

#### AboutBox

Display the About Box for the control. The About Box contains the product name and version number as well as other information about the software and Integration Objects.

AboutBox()

#### 10.7. InitializeEventCallback

InitializeEventCallback
Initialize the Event Callback.
void InitializeEventCallback()



## 11. OPCAEServer

Property	Description
ServerAddress	[Read/Write]The machine where the OPC Server is registered. Note that the host may be the IP address of the machine.
ServerProgID	[Read/Write]The OPC Server name (ProgID).
ServerStatus	[ReadOnly]The current OPC AE Server Status Structure. This Structure is an <b>IOEventServerStatus</b> object.
ServerIndex	[ReadOnly]A Boolean that defines the filter by Event.
SubscriptionsCount	[ReadOnly] the number of subscriptions.
CategoriesNumber	[ReadOnly]Returns the number of categories defined within the A&E Server once the <b>QueryEventCategories</b> has been called.
ConditionsNumber	[ReadOnly]Returns the number of conditions related to the event category once <b>QueryConditionNames</b> has been called.
SubConditionsNumber	[ReadOnly]Returns the number of subconditions related to the condition name once <b>QuerySubConditionNames</b> has been called.
SourceConditionsNumber	[ReadOnly]Returns the number of conditions related to the concerned source name once <b>QuerySourceConditionNames</b> has been called.

## 11.1. ConnectToOPCServer

	ConnectToOPCServer	
Connect to the properties.	AE Server defined by ServerProgID and ServerAddress	
int ConnectToOPCServer ()		
Return Value	Description	


hr

The operation result.

### **11.2. DisconnectfromServer**

Disconnect	
Disconnect from an AE Server that is already connected.	
int DisconnectFromServer()	
Return Value	Description
hr	The operation result.

### 11.3. CreateEventSubscription

CreateEventSubscription		
Create an event subscription object.		
OPCEventSubscription CreateEventSubscription(string SubscriptionName, bool Active, int BufferTime, int MaxSize,out int RevisedBufferTime, out int RevisedMaxSize)		
Argument	Description	
SubscriptionName	The subscription name	
Active	Initial state of the event subscription, if true, the subscription will start sending events immediately.	
BufferTime	Time in milliseconds that the server can hold back the notification to buffer multiple events (e.g. 1000 would indicate that the server can buffer events for one second and send the set of events as one network call back).	
MaxSize	The requested maximum number of events that will be sent in a single callback. A value of 0 means that there is no limit to the number of events that will be sent in a single callback.	
RevisedBufferTime	The revised buffer time that the server will actually support.	
RevisedMaxSize	The revised max size that the server will actually support.	



Return Value	Description
EventSub	The created event subscription object.

### 11.4. QueryEventCategories

QueryEventCategories			
Retrieve the available event categories for a set of event types.			
int QueryEvent	int QueryEventCategories(int iEventType)		
Argument	Description		
EventType	The event type taken from the class object OPCEventConstants (OPCEventConstants section)		
Return Value	Description		
CatgeoriesCou	nt The returned long value is the number of received event categories. If the method failed, it returns 0. Apart from the returned argument, the <b>CategoryID</b> and <b>CategoryDescription</b> Arrays will be filled respectively with category IDs and category descriptions.		

### 11.5. GetCategoryID

GetCategoryID		
Returned array of event category ID's.		
int GetCategor	int GetCategoryID(int Index)	
Argument	Description	
Index	The index of the requested category. The index should be an integer between 0 and the categories count returned by the <b>QueryEventCategories</b> function.	
Return Value	Description	
ID	The returned CategoryID given by the index.	

### **11.6. GetCategoryDescription**



GetCategoryDescription		
Returned array of event category Descriptions.		
string GetCategoryDescription(int Index)		
Argument	Description	
Index	The index of the requested category. The index should be an integer between 0 and the categories count returned by the <b>QueryEventCategories</b> function.	
Return Value	Description	
Description	The returned Category Description given by the index.	

### 11.7. QueryEventAttributes

QueryEventAttributes		
Retrieves the list of available event attributes for a given event category.		
OPCEventAttri	OPCEventAttributes QueryEventAttributes(int EventCategory)	
Argument	Description	
EventCategory	The given event category ID.	
Return Value	Description	
Object	The event attributes object : <b>OPCEventAttributes</b> (details in OPCEventAttributes Section)	

### 11.8. AckCondition

AckCondition	
Acknowledge the alarm	condition.
int AckCondition(string AcknowledgerID, string Comment, string Source, string ConditionName, long ActiveTime, int Cookie)	
Argument	Description
Argument AcknowledgerID	Description Identifying the name of acknowledging party.
Argument AcknowledgerID Source	DescriptionIdentifying the name of acknowledging party.The fully qualified source name.



ConditionName		Condition name.
ActiveTime		The time the condition went active.
ActiveTimeMilliseconds		The milliseconds of the active time.
Cookie		The cookie value of the received event.
Return Value	Description	
hr	The operation result.	

### 11.9. AckConditionActiveFileTimeAsString

	AckCondition	
Acknowledge the alarm condition using the active file time string format. This method is used in the VB6 environment development.		
int AckConditionActiveFileTimeAsString (string strAcknowledgerID, string strComment, string strSource, string strConditionName, string strActiveFileTime, int iCookie)		
Argume	nt	Description
strAcknowledg	erID	Identifying the name of acknowledging party.
strSource		The fully qualified source name.
strComment		The comment provided by the acknowledger.
strConditionName		Condition name.
strActiveFileTime		The file time the condition went active.
ActiveTimeMilliseconds		The milliseconds of the active time.
iCookie		The cookie value of the received event.
Return Value		Description
hr	The oper	ration result.

### 1.1. EnableConditionByArea

EnableConditionByArea		
Enable condition alarm events in an area.		
int EnableConditionByArea(string Area)		
Argument	Description	
Area	Enable alarm area.	





Return Value	Description
hr	The operation result.

### **1.2. EnableConditionBySource**

	EnableConditionBySource	
Enable condition alarm events for a given source.		
int EnableConditionBySource(string Source)		
Argument	Description	
Source	Enable alarm source.	
Return Value	Description	
hr	The operation result.	

### 1.3. DisableConditionByArea

	DisableConditionByArea	
Disable condition alarm events in an area.		
int DisableConditionByArea(string Area)		
Argument	Description	
Area	Disable alarm area.	
Return Value	Description	
hr	The operation result.	

### 1.4. DisableConditionBySource

DisableConditionBySource
Disable condition alarm events for a given source.
int DisableConditionBySource(string Source)



Argument	Description
Source	Disable alarm source.
Return Value	Description

### 1.5. QueryConditionNames

	QueryConditionNames	
Retrieve the lis	Retrieve the list of condition names for a specific event category.	
int QueryConditionNames(int EventCategory)		
Argument	Description	
EventCategory	The event category ID.	
Return Value	Description	
Count	The number of the received condition names. You can read the returned array of condition names by the <b>ConditionName</b> function.	

### 1.6. GetConditionName

	GetConditionName	
A property defi	ning the returned array of condition names.	
string GetConditionName(int Index)		
Argument	Description	
Index	The index of the requested condition names. The index should be an integer between 0 and the conditions count returned by the <b>QueryConditionNames</b> function.	
Return Value	Description	
CondName	Returned Condition Name.	

### 1.7. QuerySubConditionNames





	QuerySubConditionNames	
Retrieves the li	st of subcondition names for a specific condition name.	
int QuerySubConditionNames(string strConditionName)		
Argument	Description	
CondName	The condition name.	
Return Value	Description	
Count	The number of the received subcondition names. You can read the returned array of subcondition names with the <b>SubConditionName</b> function.	



### 1.8. GetSubConditionName

	GetSubConditionName
A property defin	ing the returned array of subcondition names.
string GetSubC	onditionName(int Index)
Argument	Description
Index	The index of the requested subcondition name. The index should be an integer between 0 and the subcondition's count returned by the <b>QuerySubConditionNames</b> function.
Return Value	Description
SubCondName	Returned SubCondition Name.

### 1.9. QuerySourceConditionNames

	QuerySourceConditionNames	
Retrieve the lis	Retrieve the list of condition names for a given source name.	
int QuerySubConditionNames(string strConditionName)		
Argument	Description	
Source	The source name.	
Return Value	Description	
Count	The number of the received condition names. You can read the returned array of condition names with the <b>SourceConditionName</b> function.	

### 1.10. GetSourceConditionName

GetSourceConditionName
A property defining the returned array of source condition names.
string GetSourceConditionName(int Index)



Argument		Description
Index	The inde sub <b>Qu</b> e	e index of the requested source condition name. The ex should be an integer between 0 and the ocondition's count returned by the erySourceConditionNames function.
Return Valu	e	Description
SourceCondNa	me	Returned source condition name.

### 1.11. GetConditionState

	GetConditionState
Get condition state object for a given source's condition.	
OPCConditionStat int EventCategory)	e GetConditionState(string Source, string ConditionName,
Argument	Description
	Description
Source	Source name.
Source ConditionName	Source name. Condition name.

### 1.12. GetCanFilterByEvent

GetCanFilterByEvent	
Check if the server can filter by event.	
bool GetCanFilterByEvent()	
Return Value	Description
boolean	This method returns the true if the server can filter by event and false if not

### 1.13. GetCanFilterBySeverity

Check if the server can filter by severity.



bool GetCanFilterBySeverity()	
Return Value	Description
boolean	This method returns the true if the server can filter by severity and false if not

### 1.14. GetCanFilterByCategory

GetCanFilterByCategory	
Check if the server can filter by category.	
bool GetCanFilterByCategory()	
Return Value	Description
boolean	This method returns the true if the server can filter by category and false if not

### 1.15. GetCanFilterByArea

GetCanFilterByArea	
Check if the server can filter by area.	
bool GetCanFilterByArea()	
Return Value	Description
boolean	This method returns the true if the server can filter by area and false if not

### 1.16. GetCanFilterBySource

GetCanFilterBySource	
Check if the server can filter by source.	
bool GetCanFilterBySource()	
Return Value	Description
boolean	This method returns the true if the server can filter by source and false if not



### 1.17. CreateBrowser

CreateBrowser	
Create an event browser object for the server.	
OPCEventBrowser CreateBrowser()	
Return Value	Description
Browser	This method returns the OPCEventBrowser object related to the connected server.

### 1.18. GetCategoryDescriptionfromID

GetGategoryDescriptionfromID	
Return the category description for a given category ID once the QueryEventCategories has been called	
string GetCategoryDescriptionfromID(int ID)	
Return Value	Description
catDsc	This method returns the event category description related to a given category ID.

### 1.19. GetCategoryDescriptionfromID

GetGategoryDescriptionfromID	
Return the category description for a given category ID once the QueryEventCategories has been called	
string GetCategoryDescriptionfromID(int ID)	
Return Value	Description
catDsc	This method returns the event category description related to a given category ID.

### 1.20. GetCategoryIDfromDescription

#### GetCategoryIDfromDescription

Return the category ID for a given category description once the **QueryEventCategories** has been called



int GetCategoryIDfromDescription(string CatDesc)	
Return Value	Description
ID	This method returns the event ID related to a given category description.

### 2. OPCEventSubscription

Using the following properties, the user can set up filtering options for event subscription and retrieve the events.

Property	Description
SubscriptionName	(Read Only) The subscription name
FilterEventType	(Read/Write) A Bit mask indicating the selected event types that should be sent to this subscription, built by combining values from the class object <b>OPCEventConstants</b> (OPCEventConstants section). Only events satisfying the criterion "FilterEventType" will be returned.
FilterHighSeverity	(Read/Write) The Highest Severity limit for this subscription (Between 1 and 1000) and must be equal to or greater than FilterLowSeverity.
FilterLowSeverity	(Read/Write) The Low Severity limit for the subscription (Between 1 and 1000) and must be equal to or less than FilterHighSeverity.
FilterCategoriesCount	(Read-only) The number of category IDs. If 0, all categories are selected.
FilterSourcesCount	(Read-only) The number of sources. If 0, all sources are selected.
FilterAreasCount	(Read-only) The number of areas. If 0, all areas are selected.
FilterCategories	(Read-only) Array of Category ID's (see QueryEventCategories method) that should be sent to the event subscription object.
FilterAreas	(Read-only) List of area names.
FilterSources	(Read-only) List of source names.

### 2.1. GetSubscriptionState



#### GetSubscriptionState

Get subscription state information.

void GetSubscriptionState(out bool Active, out int BufferTime, out int MaxSize, out int hClientSubscription)

Argument	Description
Active	When Active is true, the event subscription is active and it sends the OnIncomingEvent notifications.
BufferTime	Time in milliseconds that the server can hold back the notification to buffer multiple events.
MaxSize	The maximum number of events to buffer before sending events.
hClientSubscription	The client handles for the subscription.

### 2.2. SetSubscriptionState

SetSubscriptionState		
Modify the subscription state information.		
void SetSubscriptionState(bool Active, int BufferTime, int MaxSize, int hClientSubscription, out int RevisedBufferTime, out int RevisedMaxSize)		
Argument	Description	
Active	When Active is true, the event subscription is active and it sends the OnEvent notifications.	
BufferTime	Time in milliseconds that the server can hold back the notification to buffer multiple events.	
MaxSize	The maximum number of events to buffer before sending events.	
hClientSubscription	The client handles for the subscription.	
RevisedBufferTime	The revised buffer time that the server will support.	
RevisedMaxSize	The revised maximum size that the server will support.	

#### 2.3. Refresh

	Refresh
Refresh the subscription event.	
void Refresh()	



#### 2.4. CancelRefresh

CancelRefresh
Cancel the Refresh of the subscription event.
void CancelRefresh()

### 2.5. Activate

	Activate
Activate the subscription event.	
void Activate()	

### 2.6. Deactivate

Deactivate
Deactivate the subscription event.
void Deactivate()

### 2.7. SelectAllAttributesForAllCatgories

SelectAllAttributesForAl	Catgories
	Joargenee

Selects all event attributes for all categories to be returned.

void SelectAllAttributesForAllCatgories()

### 2.8. GetReturnedEventAttributes

GetReturnedEventAttributes	
Selects all event attributes for all categories to be returned.	
OPCEventAttributes GetReturnedEventAttributes(int EventCategory)	
Argument	Description
Argument EventCategory	Description The event Category ID for which the caller wishes to get the returned attributes.



OPCEventAttributes	An OPCEventAttributes object.
Object	

### 2.9. SelectReturnedEventAttributes

GetReturnedEventAttributes	
Selects event attributes for a specific category to be returned.	
<pre>void SelectReturnedEventAttributes(int EventCategory, int[] AttributeIDs, int Count)</pre>	
Argument	Description
Argument EventCategory	Description The event Category ID.
Argument EventCategory AttributeIDs	Description           The event Category ID.           The array of attribute ID's.

### 2.10. GetFilter

GetFilter
Get OPC Filter options for this event subscription.
void GetFilter()

### 2.11. ApplyFilter

ApplyFilter
Set up the filtering options for this event subscription.
void ApplyFilter()

### 2.12. AddFilterSource

AddFilterSource	
Add a source name to the array of filtering source names.	
void AddFilterSource(string Source)	
Argument Description	



Source

The source name added to the filter source list.

### 2.13. AddFilterArea

AddFilterArea		
Add an area name to the array of filtering area names.		
void AddFilterArea(string Area)		
Argument	Description	
Area	The area name added to the filter areas list.	

### 2.14. AddFilterCategory

AddFilterCategory		
Add a category ID to the array of filtering categories IDs.		
void AddFilterCategory(int CategoryID)		
Argument	Description	
CategoryID	The category ID	

### 2.15. RemoveFilterSource

RemoveFilterSource			
Remove a source name from the source list.			
void RemoveFilte	void RemoveFilterSource(int iIndex)		
Argument	Description		
Index	A given index that references the index of the source in the list to be removed. The index is between 0 and the source count $-1$ .		

### 2.16. RemoveFilterArea

RemoveFilterArea
Remove the area name from the area list.
void RemoveFilterArea(int Index)



Argument	Description	
Index	A given index that references the index of the area in the list to be removed. The index is between 0 and the area count $-1$ .	
Example		
RemoveFilterArea (1)		

### 2.17. RemoveFilterCategory

RemoveFilterCategory		
Remove the category ID from the category list.		
void RemoveFilterCategory(int Index)		
Argument	Description	



### 2.18. ClearFilterSources

|--|--|

Clear the source list.

void ClearFilterSources()

### 2.19. ClearFilterAreas

	ClearFilterAreas
Clear the area list.	
void ClearFilterAreas()	

### 2.20. ClearFilterCategories

	ClearFilterCategories
Clear the category list.	
void ClearFilterCategories()	

### 3. OPCConditionState

Property	Description
Acknowledged	(Read-only) True if the condition is acknowledged.
AcknowledgerID	(Read-only) The ID string for the last acknowledger of the condition.
Active	(Read-only) True if the condition is active.
ActiveSubCondition	(Read-only) The name of the active subcondition.



Comment	(Read-only) The comment string provided by the last acknowledger of the condition.
Quality	(Read-only) The OPC quality of the condition source.
LastAckTime	(Read-only) The time the condition was last acknowledged.
State	(Read-only) The condition state.
ActiveSubConditionDefinition	(Read-only) The active subcondition definition.
ActiveSubConditionDescriptio	(Read-only) The active subcondition description.
ActiveSubConditionSeverity	(Read-only) The active subcondition severity.
SubConditionLastActiveTime	(Read-only) The time that the subcondition last became active.
ConditionLastActiveTime	(Read-only) The time that the condition last became active.
ConditionLastInactiveTime	(Read-only) The time that the condition last became inactive.
State	(Read-only) The condition state.
SubConditionsCount	(Read-only) Number of subconditions.
QualityString	(Read-only) A string representation of quality string.
SubConditionDefinitions	(Read-only) The definitions of the condition's subconditions.
SubConditionDescriptions	(Read-only) The descriptions of the condition's subconditions.
SubConditionNames	(Read-only) The names of the condition's subconditions.
SubConditionSeverities	(Read-only) The severities of the condition's subconditions.

### 3.1. GetQuality

GetQuality
Return the quality value.
int GetQuality()



#### 3.2. GetState

GetState
Return the state value.
int GetState()

### 3.3. GetActiveSubConditionDefinition

GetActiveSubConditionDefinition

Return the active subcondition definition string.

string GetActiveSubConditionDefinition()

### 3.4. GetActiveSubConditionDescription

G	etAct	iveS	ubCon	ditionl	Descripti	on	

Return the active subcondition description string

string GetActiveSubConditionDescription()

### 3.5. GetActiveSubConditionSeverity

**GetActiveSubConditionSeverity** 

Return the active subcondition severity value.

string GetActiveSubConditionSeverity()

### 3.6. GetSubConditionLastActiveTime

#### **GetSubConditionLastActiveTime**

Return the subcondition last active time datetime.

DateTime GetSubConditionLastActiveTime()



### 3.7. GetConditionLastActiveTime

#### GetConditionLastActiveTime

Return the condition last active time datetime

DateTime GetConditionLastActiveTime()

### 3.8. GetConditionLastInactiveTime

**GetConditionLastInactiveTime** 

Return the condition last inactive time datetime

DateTime GetConditionLastInactiveTime ()

### 3.9. GetStateString

	GetStateString
Return the state string.	
string GetStateString()	

### 3.10. GetSubConditionsCount

GetSubConditionsCount
Return the subConditions count.
int GetSubConditionsCount()

### 3.11. GetSubConditionDefinitions

	GetSubConditionDefinitions
Return the subCo	ndition definition string for a given index.
string GetSubCor	nditionDefinitions(int Index)
Argument	Description



### 3.12. GetSubConditionDescriptions

	GetSubConditionDescriptions
Return the subCo	ndition description string for a given index.
string GetSubCor	nditionDescriptions(int Index)
Argument	Description
Index	A given index that references the index of the sub condition descriptions in the list to be returned. The index is between 0 and the sub condition descriptions count –1.

### 3.13. GetSubConditionSeverities

	GetSubConditionSeverities
Return the subCo	ndition severity value for a given index.
int GetSubCondit	ionSeverities(int Index)
Argument	Description
Index	A given index that references the index of the sub condition severity in the list to be returned. The index is between 0 and the sub condition severities count $-1$ .

### 3.14. GetSubConditionNames

	GetSubConditionNames
Return the subCo	ndition name string for a given index.
string GetSubCor	nditionNames(int Index)
Argument	Description

### 3.15. GetQualityString

GetQualityString



Return the quality string.

string GetQualityString()

### 3.16. GetSubConditionLastActiveTimeAsString

GetSubConditionLastActiveTimeAsString

Return the sub condition last active time string.

string GetSubConditionLastActiveTimeAsString()

### 3.17. GetConditionLastActiveTimeAsString

GetConditionLastActiveTimeAsString

Return the condition last active time string.

string GetConditionLastActiveTimeAsString()

### 3.18. GetSubConditionLastInactiveTimeAsString

GetSubConditionLastInactiveTimeAsString

Return the sub condition last inactive time string.

string GetConditionLastInactiveTimeAsString ()

### 3.19. GetLastAckTimeAsString

#### GetLastAckTimeAsString

Return the last acknowledgment time string.

string GetLastAckTimeAsString()

### 4. EventStruct



Property	Description
AckRequired	(Read-only) A Boolean that defines whether the alarm condition requires Acknowledgment or not.
ActiveTime	(Read-only) The time the condition alarm went active.
EventTime	(Read-only) The time when the event occurred.
ConditionName	(Read-only) The condition name (condition events).
SubConditionName	(Read-only) The subcondition name (condition events).
EventCategory	(Read-only) Event category ID.
Cookie	
Message	(Read-only) The alarm message.
NewState	(Read-only) Bit mask that indicates the condition's current state, based on the class object <b>OPCEventConstants</b> (OPCEventConstants section)
Quality	(Read-only) The quality value for condition events.
Severity	(Read-only) The severity value for condition events.
Source	(Read-only) The source item that caused the event to occur.
EventType	(Read-only) The event type. The allowable values are values from the class object <b>OPCEventConstants</b> (OPCEventConstants section)
ActorID	(Read-only) The acknowledger name (condition events).
ChangeMask	(Read-only) Bit mask indicating which items have changed. This property is based on the class object <b>OPCEventConstants</b> (OPCEventConstants section)
EventTimeMilliseconds	(Read-only) The milliseconds in the event time.
ActiveTimeMilliseconds	(Read-only) The milliseconds in the active time.
EventAttributesCount	(Read-only) The number of event attributes.
EventAttributes	(Read-only) Array of event attributes' values.
EventAttributesAsString	(Read-only) Array of event attributes' values as string.

### 4.1. GetEventAttributes

#### GetEventAttributes

Return the sub condition last active time string.



object GetEventA	.ttributes(int Index)
Argument	Description
Index	A given index that references the index of the event attribute object in the list to be returned. The index is between 0 and the event attributes count $-1$ .

### 4.2. GetEventAttributesAsString

	GetEventAttributesAsString
Return the sub co	ondition last active time string.
string GetEventAt	tributesAsString(int Index)
Argument	Description
<i>i</i> gamon	Description

### 4.3. GetStrQuality

GetStrQuality
Return the quality string.
string GetStrQuality()

### 4.4. GetEventTimeAsString

GetEventTimeAsString	
Return the event time string.	
string GetEventTimeAsString()	

### 4.5. GetActiveTimeAsString

#### GetActiveTimeAsString

Return the active time string.



string GetActiveTimeAsString()

### 5. OPCServerStatus

Property	Description
ServerState	(Read-only)The current status of the AE Server.
	Allowable values are given by the enumeration : IO_OPCAE_SERVERSTATE_CONSTANTS (see the following table).
CurrentTime	(Read-only)The current time of the OPC AE Server.
StartTime	(Read-only)The time when the server process started.
LastUpdateTime	(Read-only)The last updated time of the server process.
MajorVersion	(Read-only)The major version identification of the historian.
MinorVersion	(Read-only)The minor version identification of the historian.
BuildNumber	(Read-only)The build number identification of the historian
VendorInfo	(Read-only)The vendor information for the OPC Server.
CurrentTimeMilliseconds	(Read-only)The milliseconds of the current time of the OPC AE Server.
StarttimeMilliseconds	(Read-only)The milliseconds of the time when the server process started.
LastUpdateTimeMilliseconds	(Read-only)The last updated time in milliseconds.

### 4.1. GetServerStatusString

GetServerStatusString	
Return the server status string.	
string GetServerStatusString()	



### 4.2. GetStartTimeString

GetStartTimeString	
Return the start time string.	
<pre>string GetStartTimeString()</pre>	

### 4.3. GetLastUpdateString

GetLastUpdateString	
Return the last update time string.	
string GetLastUpdateString()	

### 4.4. GetCurrentTimeString

GetCurrentTimeString	
Return the current time string.	
string GetCurrentTimeString()	

### 6. OPCEventBrowser

Property	Description
SelectedArea	(Read-only) The selected area in the browser.
SelectedSource	(Read-only) The selected source in the browser.

### 6.1. ShowBrowserDialog

ShowBrowserDialog
This method allows users to access a browser dialog to browse all address space and to select the source or area you want.
bool ShowBrowserDialog()



### 7. OPCEventAttributes

Property	Description
AttributesCount	(Read-only) The number of attributes in the structure.
AttributeIDs	(Read-only) Array of the attributes' IDs.
AttributeDescriptions	(Read-only) Array of the attributes' Descriptions.
AttributeVarTypes	(Read-only) Array of the attributes' Data Types.

### 7.1. GetAttributeIDs

GetAttributeIDs	
Return the attribu	te ID.
int GetAttributeID	s (int Index)
Argument	Description
Index	A given index that references the index of the event attribute ID in the list to be returned. The index is between 0 and the event attributes count $-1$ .

### 7.2. GetAttributeDescriptions

GetAttributeDescriptions	
Return the attribu	te description.
string GetAttribute	Descriptions (int Index)
Argument	Description
Index	A given index that references the index of the event attribute

### 7.3. GetAttributeVarTypes

	GetAttributeVarTypes
the attribute vartype	

Return



int GetAttributeVarTypes (int Index)		
Argument	Description	
Index	A given index that references the index of the event attribute vartypes in the list to be returned. The index is between 0 and the event attributes vartypes count –1.	

### 7.4. GetVarTypeString

GetVarTypeString		
Return the attribute vartype string		
string GetVarTypeString (int Index)		
Argument	Description	
Index	A given index that references the index of the event attribute vartype string in the list to be returned. The index is between 0	

### 8. OPCEventFilter

Property	Description	
EventType	(Read\Write) the event type	
LowSeverity	(Read\Write) the low severity	
HighSeverity	(Read\Write) the high severity	
FilterCategoriesCount	(Read\Write) the filter categories count	
FilterAreasCount	(Read\Write) the filter areas count	
FilterSourcesCount	(Read\Write) the filter sources count	

### 8.1. SetEventCategories

SetEventCategories		
Set the event category of a given index		
void SetEventCategories(int Index, int NewVal)		
Argument	Description	
Index	A given index that references the index of the event category ID in the list to be modified. The index is between 0 and the event	



	categories count –1.
NewVal	The new value

### 8.2. SetFilterAreas

SetFilterAreas		
Set the area of a given index		
void SetFilterAreas(int Index, string NewVal)		
Argument	Description	
Index	A given index that references the index of the area in the list to be modified. The index is between 0 and the event areas count $-1$ .	
NewVal	The new value	

### 8.3. GetFilterAreas

GetFilterAreas		
Return the area of a given index		
string GetFilterAreas(int Index)		
Argument	Description	
Index	A given index that references the index of the event area in the list to be returned. The index is between 0 and the event areas count $-1$ .	

### 8.4. GetEventCategories

GetEventCategories		
Return the event category of a given index		
int GetEventCategories(int Index)		
Argument	Description	
Index	A given index that references the index of the event categories ID in the list to be returned. The index is between 0 and the event categories ID count $-1$ .	



### 8.5. GetFilterSources

GetFilterSources		
Return the source of a given index		
string GetFilterSources(int Index)		
Argument	Description	
Index	A given index that references the index of the event source in the list to be returned. The index is between 0 and the event areas count $-1$ .	

### 8.6. SetFilterSources

SetFilterSources		
Set the source of a given index		
void SetFilterSources(int Index, string NewVal)		
Argument	Description	
Index	A given index that references the index of the source in the list to be modified. The index is between 0 and the event sources count $-1$ .	
NewVal	The new value	

### 9. OPCEventConstants

### 9.1. OPC FILTER CONSTANTS

Valu	es
OPC_FILTER_BY_EVENT	
OPC_FILTER_BY_CATEGORY	
OPC_FILTER_BY_SEVERITY	
OPC_FILTER_BY_AREA	
OPC_FILTER_BY_SOURCE	

### 9.2. OPC EVENT TYPES CONSTANTS

Values





OPC\_SIMPLE\_EVENT OPC\_TRACKING\_EVENT OPC\_CONDITION\_EVENT

OPC\_ALL\_EVENTS

### 9.3. OPC CONDITION STATE CONSTANTS

	Values	
OPC_CONDITION_ENABLED		
OPC_CONDITION_ACTIVE		
OPC_CONDITION_ACKED		

### 9.4. OPC CHANGE CONSTANTS

·	Values
OPC_CHANGE_ACTIVE_STATE	
OPC_CHANGE_ACK_STATE	
OPC_CHANGE_ENABLE_STATE	
OPC_CHANGE_QUALITY	
OPC_CHANGE_SEVERITY	
OPC_CHANGE_SUBCONDITION	
OPC_CHANGE_MESSAGE	
OPC_CHANGE_ATTRIBUTE	

# 5. Deploying the Event Client .Net ActiveX in Microsoft Visual Basic 6.0

### 5.1. Create a Standard EXE

Run the Microsoft Visual Basic 6.0 and select a Standard EXE to create a new project as shown below.



🍖 New Project					X
	Micros	oft ual B	asi	C	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
New Existing	Recent				1
Standard EXE	ActiveX EXE	ActiveX DLL	ActiveX Control	VB Application Wizard	* III
VB Wizard Manager	ActiveX Document Dll	Activex Document Exe	Addin	Data Project	
P= 6	P= 💊	P= 6			-
				<u>O</u> pen Cancel	-
				<u>H</u> elp	
Don't show thi	s dialog in the f <u>u</u> t	ture			

Figure 114: Create a VB6 Standard EXE

### 5.2. Add the OPC Event Client .Net reference

1. Select the Project menu item and click on References





Figure 115: Select Projet Reference

 Click on the Browse button and select the OPC Event Client tlb file path (.:\Program Files (x86)\Integration Objects\Integration Objects' OPC AE ActiveX\Bin\DotNet\ OPCEventClientActiveX)



References - Project1	×
<u>A</u> vailable References:	ОК
✓ Visual Basic For Applications ✓ Visual Basic runtime objects and procedures	Cancel
Visual Basic objects and procedures	Browse
AccessControl 1.0 Type Library     AccessibilityCplAdmin 1.0 Type Library     Active DS Type Library	
ActiveMovie control type library ActiveX DLL to perform Migration of MS Repository V: AdHocReportingExcelClientLib	Help
AgControl 5.1 Type Library AP Client 1.0 HelpPane Type Library AP Client 1.0 Type Library	
۲	
OLE Automation	
Location: C:\Windows\system32\stdole2.tlb	
Language: Standard	

Figure 116: Browse the OPC AE Net Logger Path

3. Select the "IntegrationObjects.OPCEventClientDotNetActiveX.tlb" and click on the *Open* button.

Add Reference	×
Look in: DPCEventClientActiveX 🔹 🗲 🔁	) 💣 🎟 🗸
Name	Date modified
IntegrationObjects.Logger.SDK.dll	04/04/2016 09:36
Network Content Client Dot Net Active X.dll	30/01/2017 10:04
IntegrationObjects.OPCEventClientDotNetActiveX.tlb	30/01/2017 10:20
S IntegrationObjects.OPCNetClientSDK.dll	01/07/2016 12:47
🚳 License.dll	20/01/2017 15:16
<	۱.
File <u>n</u> ame: IntegrationObjects.OPCEventClientDotNetActive	<u>O</u> pen
Files of type:         Type Libraries (*.olb;*.tlb;*.dll)	Cancel
	Help

Figure 117: Select the type library (.tlb) file

4. Once the IntegrationObject\_OPCEventClientDotNetActiveX reference is checked, click on OK



References - Project1	×			
<u>A</u> vailable References:	ОК			
IIS Installed Versions Manager Interface	Cancel			
Imapikor 1.0 Type Library IMContact 1.0 Type Library IMEAPI JK 1.0 Type Library	Browse			
InfoPath Property Promotion Outlook UI 1.0 Type Lib InsRepIM 1.0 Type Library InstallShield Professional Setup Kernel D9.0 InstallShield Script 1.0 Type Library InstallShield Setup Kernel InstallShield Setup Kernel InstallShield Windows Installer Setup Kernel 1.0 Type IntegrationObjects_OPCAELoggerDotNetActiveX	Help			
IntegrationObjects OPCEventClientDotNetActiveX III III III III III III III III III I				
_IntegrationObjects_OPCEventClientDotNetActiveX				
Location: C:\Users\Admin\Desktop\ActiveX\IntegrationObjects.OPCEver				

Figure 118: Check the OPC Event Client Net ActiveX Reference

## 5.3. Add the OPC Event Client Net ActiveX Component to the Toolbar

5. Select the *Project* menu item and click on *Components*




Figure 119: Select Projet Components

6. Check the IntegrationObject\_OPCEventClientDotNetActiveX component then click on the *OK* button



Components	×
Controls Designers Insertable Objects HHActiveX 1.0 Type Library HybridAppControls iFIX Color Button Control module iFix Key Macro Editor Keyboard Control	
Integration Objects' OPC AE Controls Integration Objects' OPC Event Client ActiveX IntegrationObjects_OPCAELoggerDotNetActive IntegrationObjects_OPCEventClientDotNetActi IntegrationObjects_OPCEventClientDotNetActive IntegrationIntegrative Integrative	
IOOPCActive ActiveX Control module IStudio Active Designer Controls KM Rdp Protocol Provider 1.0 Type Library LayoutDTC 1.0 Type Library	
IntegrationObjects_OPCEventClientDotNetActive Location: C:\Windows\system32\mscoree.dll	eX
Close	Cancel <u>Apply</u>

Figure 120: Select the OPC Event Net ActiveX Component

7. Select the IOEventClientDotNetCtrl already added in the toolbar and draw the control in the form as shown below



### OPC AE ActiveX User Manual



### Figure 121: Add the OPC Event Client Net ActiveX Component to the Form



# **CONFIGURING DCOM**

In order to retrieve data from OPC Server(s) in real time, Integration Objects' OPC AE ActiveX can be used in different configurations, including local and distributed configurations. In a local configuration, the toolkit and OPC Server(s) all run on the same computer. In that case, the installation process does not need any specific settings. In a distributed configuration, these components are executed on two or more computers cooperatively: Integration Objects' OPC AE ActiveX software initially resides on a remote computer (Client Computer) and uses the DCOM mechanism to directly access server(s). To enable this functionality, some additional settings are needed on both the remote server and the local client computer. This section is intended to provide general guidance on proper DCOM Config Utility settings for computers on which the OPC AE Controls and OPC server(s) are running.

### 1. Client Side DCOM Configuration

**<u>Step 1</u>**: Once you log in as an Administrator, setup Client machine with these instructions:

1. Choose the Run Option from the Windows' Start menu and type DCOMCNFG then click *OK* to run it. When you first launch the utility, it will look like this:



Distributed COM Configuration Properties	<u>? ×</u>
Applications Default Properties Default Security Default Protocols	
Applications: APE Worker Provider APIDeclarationLoader.AddInDesigner Bitmap Image Defrag FAT engine Defrag NTFS engine Document de Systran Professional Premium EnumTelnetClientsSvr Class Foxhtmlhelp Class HDATestSvr HTML Application HtmlHelp Class IisServiceControl Class	*
Integration Objects OPC For PI Internet Explorer(Ver 1.0) Logical Disk Manager Administrative Service Logical Disk Manager Remote Client Machine Debug Manager	T
OK Cancel Ap	ply

Figure 122: Distributed COM Configuration

- 2. DCOM Configuration Properties-Default Properties Tab:
  - a. The Enable Distributed COM on this computer MUST be checked.
  - b. The Default Authentication Level should be set to Connect.
  - c. The Default Impersonation Level should be set to Identity.



Distributed COM Configuration Properties	? ×
Applications Default Properties Default Security Default Protocols	
Enable Distributed COM on this computer	
Enable COM Internet Services on this computer	
Default Distributed COM communication properties	
The Authentication Level specifies security at the packet level.	
Default A <u>u</u> thentication Level:	
Connect	
The Impersonation Level specifies whether applications can determine who is calling them, and whether the application can do operations using the client's identity. Default Impersonation Level: Identify	
OK Cancel App	dy

Figure 123: DCOM Default Properties

3. DCOM Configuration Properties-Default Security Tab:

It is on this tab that you can tell the operating system who is allowed to access the applications deploying the OPC AE Controls from remote OPC servers. Default Access Permission is the only setting we are concerned with on the client side of this tab. On the Default Access Permissions Dialog, you will set who (users whose remote OPC servers are running under) will have the ability to make callbacks to this machine when subscription based reads are being done. No changes are normally required on Default Launch Permissions and Default Configuration Permissions dialogs.



istributed COM Configuration Properties	X
Applications Default Properties Default Security Default Protocols	
Default Acc <u>e</u> ss Permissions	
You may edit who is allowed to access applications that do not provide their own settings	
Edit Default	
Default Launch Permissions	
You may edit who is allowed to launch applications that do not provide their own settings.	
Edit Default	
Default Configuration Permissions	
You may edit the list of users that are allowed to modify OLE class configuration information. This includes installing new OLE servers and adjusting the configuration of existing OLE servers.	
Edit Default	
OK Cancel Apply	

Figure 124:DCOM Default Security Tab

4. DCOM Configuration Properties-Default Protocols Tab:

On this tab you set which of the installed network protocols to use for DCOM on the client computer. You should use Connection-oriented TCP/IP.



Distributed COM Configuration Properties
Applications Default Properties Default Security Default Protocols
DCOM Protocols
译 Datagram UDP/IP 译 Datagram IPX
Connection-oriented TCP/IP
Connection-oriented NetBEUI
Add Remove Move Up Move Down Properties
The set of network protocols available to DCOM on this machine. The ordering of the protocols reflects the priority in which they will be used, with the top protocol having first priority.
OK Cancel Apply

Figure 125 : DCOM Default Protocols

<u>Step 2:</u> You need to register your OPC Server on the client computer by indicating its location on a named remote machine.

Depending on the client environment, there are two ways of registering your OPC server on your client machine. Here are two methods:

- Prepare and apply a customized .reg file on the client computer (See Microsoft registry documentation for details). We recommend this method only for experienced users with Windows Registry. You have to export the entries of your OPC server from the server machine registry to the client machine registry.
- 2. Alternatively, install and configure your OPC Server on the client computer. This action self-registers the server in the System Registry. This is the easiest way for an automatic registration.

Then, use the following steps to verify that the OPC server machine is properly delegated:

- a. On the client machine, run the DCOM Config Utility (Dcomcnfg.exe).
- b. Select your OPC server from the Applications tab and choose Properties.
- c. On the General tab, be sure that there is an entry for Remote Computer and that the remote computer name is correct.
- d. If the computer name is incorrect, select the Location tab.



e. Make sure the Run application on the following computer setting is checked. In the Dialog box beneath this selection, type in the correct computer name for your OPC server (see the figure below).

Integration Objects OPC	For PI Properties
General Location Security Identity Endpoints	
General properties of this DCOM application	
Application name:	Integration Objects OPC For PI
Application type:	local server
Authentication Level:	Default
Local path:	C:\Users\temp\code\svr\Debug\opc_exe.exe
	OK Cancel Apply

Figure 126: General properties of the selected OPC server

You can also use the following steps to verify the remote computer name by using the Windows Registry:

- 1. Run RegEdit.exe.
- 2. The remote server name is specified in the following registry key:
- 3. HKEY\_CLASSES\_ROOT\AppID\{The CLSID of the OPC server}\ RemoteServerName



## **10. Server Side DCOM Configuration**

There are 2 areas you will need to setup:

<u>Step 1</u>: Follow these instructions to make default DCOM Configuration for your OPC Server Computer.

- 1. Launch the DCOM Config Utility on the computer your target OPC Server is running.
- 2. Configure the Default Properties Tab as you did on the Client side.
- 3. DCOM Configuration Properties-Default Security Tab:

This tab has the most settings to be made. On this tab, specify for the operating system who is allowed to access OPC servers on this machine (Default Access Permissions), who is allowed to launch OPC Servers on this machine (Default Launch Permissions), and who is allowed to configure OPC Servers on this machine (Default Configuration Permissions).

4. DCOM Configuration Properties-Default Security Tab-Default Access Permissions Dialog:

In the grant access to users and groups, click *Add* on the right of the dialog and you will be presented with another dialog that allows you to browse and select users on the local machine and domain (if applicable and logged into a domain).

5. DCOM Configuration Properties-Default Security Tab-Default Launch Permissions Dialog:

Here, you can define who can actually start your OPC server on this computer. Adding of users/groups is done the same way as was done for Access Permissions.

- 6. DCOM Configuration Properties-Default Security Tab-Default Configuration Permissions Dialog: If you are setting up DCOM for the first time, it is not recommended to change the settings.
- 7. Configure the Default Protocols Tab as you did on the Client side.

**<u>Step 2</u>**: To make DCOM settings that are specific to your OPC Server, go to the Application Tab in DCOM Config and browse until you find the OPC Server of your choice. Highlight it and either double click it or click the Properties to enter the server specific settings.

1. On the General Tab, we recommend that you leave the Authentication Level as **Default**.



Integration Objects OPC	For PI Properties
General Location Secu	urity Identity Endpoints
General properties of th	nis DCOM application
Application name:	Integration Objects OPC For PI
Application type:	local server
Authentication Level:	Default
Local path:	C:\Users\temp\code\svr\Debug\opc_exe.exe
	OK Cancel Apply

Figure 127 :DCOM Server Configuration

2. On the Location Tab, make sure that the *Run Application* on this computer is the ONLY box that is checked.



Integration Objects OPC For PI Properties	<u>?</u> ×
General Location Security Identity Endpoints	
The following settings allow DCDM to locate the correct application. If you make more than one selection, then D applicable one. Client applications may override your sel	computer for this )COM uses the first ections.
Run application on the computer where the data is logonal strength of the s	ocated
Run application on this computer	
Run application on the following computer:	
	Browse
	A Apple
	Nutrie and an and an

Figure 128:Location configuration of the selected OPC server

3. On the Security Tab, we suggest that you select "Use Default access permissions". This means that users/groups shown under the *Default Security Tab* in the DCOM Config utility will have access to connect to this specific OPC server. If you choose to use the custom permissions to override the defaults, you must specify which users/groups you wish to grant access to. We also suggest that you use the Default Launch permissions. The same rules apply about using custom launch permissions here as they do for custom access permissions.



Integration Objects OPC For PI Properties	? ×
General Location Security Identity Endpoints	
<ul> <li>Use default access permissions</li> <li>Use custom access permissions</li> <li>You may edit who can access this application.</li> </ul>	
<ul> <li>Use default launch permissions</li> <li>Use custom launch permissions</li> <li>You may edit who can launch this application.</li> </ul>	
<ul> <li>Use default configuration permissions</li> <li>Use custom configuration permissions</li> <li>You may edit who can change the configuration information for this application.</li> </ul>	
OK Cancel A	pply

Figure 129: Configuration of DCOM security properties

- 4. On the *Identity* Tab, specify under which user account you want the OPC server to run. For some cases, this is one of the most important settings for the OPC server. The answer is very dependent on how you will be using your system.
- 5. No changes are required on the *Endpoints* Tab.



For additional information on this guide, questions or problems to report, please contact:

### Offices

- Americas: +1 713 609 9208 - Europe-Africa-Middle East:
  - +216 71 195 360

#### Email

- Support Services: customerservice@integrationobjects.com
- Sales: sales@integrationobjects.com

To find out how you can benefit from other Integration Objects products and custom-designed solutions, please visit us on the Internet:

#### Online

- www.integrationobjects.com