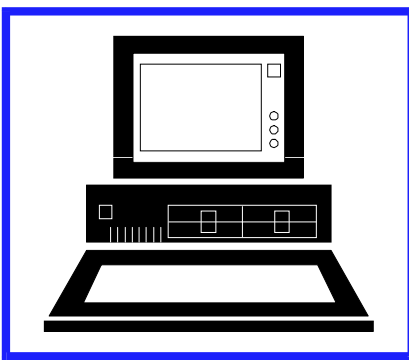
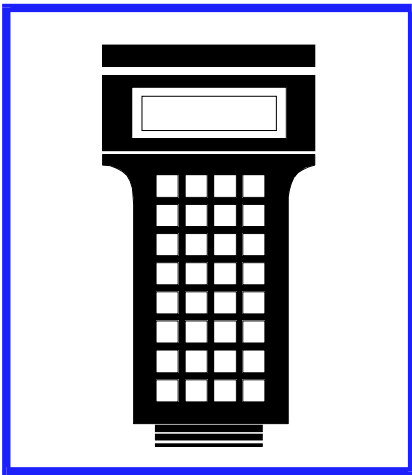
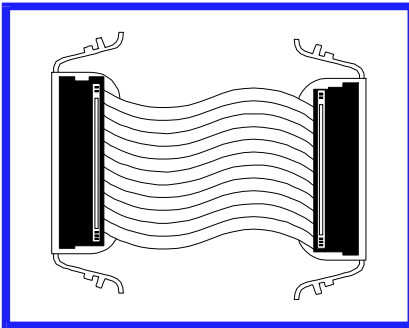
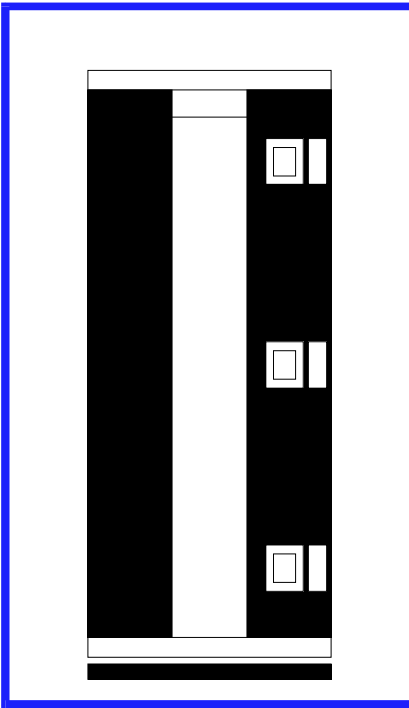
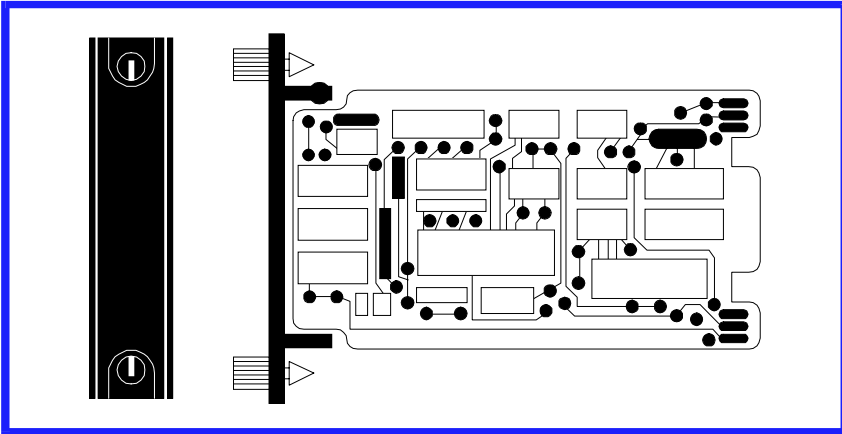
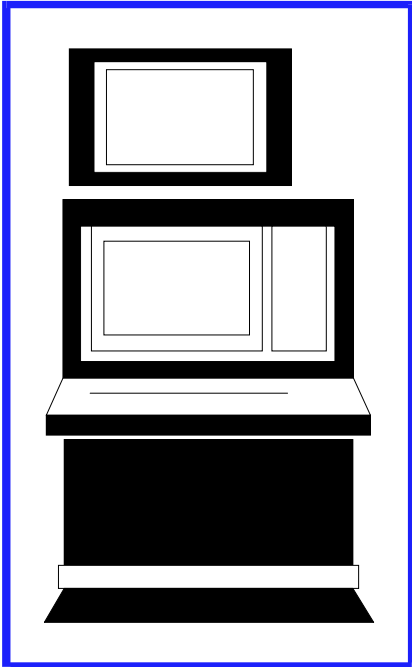


ODS/ODC

Instruction

Open Data Server/Client



WARNING notices as used in this instruction apply to hazards or unsafe practices that could result in personal injury or death.

CAUTION notices apply to hazards or unsafe practices that could result in property damage.

NOTES highlight procedures and contain information that assists the operator in understanding the information contained in this instruction.

WARNING

INSTRUCTION MANUALS

DO NOT INSTALL, MAINTAIN, OR OPERATE THIS EQUIPMENT WITHOUT READING, UNDERSTANDING, AND FOLLOWING THE PROPER **Elsag Bailey** INSTRUCTIONS AND MANUALS; OTHERWISE, INJURY OR DAMAGE MAY RESULT.

RADIO FREQUENCY INTERFERENCE

MOST ELECTRONIC EQUIPMENT IS INFLUENCED BY RADIO FREQUENCY INTERFERENCE (RFI). CAUTION SHOULD BE EXERCISED WITH REGARD TO THE USE OF PORTABLE COMMUNICATIONS EQUIPMENT IN THE AREA AROUND SUCH EQUIPMENT. PRUDENT PRACTICE DICTATES THAT SIGNS SHOULD BE POSTED IN THE VICINITY OF THE EQUIPMENT CAUTIONING AGAINST THE USE OF PORTABLE COMMUNICATIONS EQUIPMENT.

POSSIBLE PROCESS UPSETS

MAINTENANCE MUST BE PERFORMED ONLY BY QUALIFIED PERSONNEL AND ONLY AFTER SECURING EQUIPMENT CONTROLLED BY THIS PRODUCT. ADJUSTING OR REMOVING THIS PRODUCT WHILE IT IS IN THE SYSTEM MAY UPSET THE PROCESS BEING CONTROLLED. SOME PROCESS UPSETS MAY CAUSE INJURY OR DAMAGE.

NOTICE

The information contained in this document is subject to change without notice.

Elsag Bailey, its affiliates, employees, and agents, and the authors and contributors to this publication specifically disclaim all liabilities and warranties, express and implied (including warranties of merchantability and fitness for a particular purpose), for the accuracy, currency, completeness, and/or reliability of the information contained herein and/or for the fitness for any particular use and/or for the performance of any material and/or equipment selected in whole or part with the user of/or in reliance upon information contained herein. Selection of materials and/or equipment is at the sole risk of the user of this publication.

This document contains proprietary information of Elsag Bailey, Elsag Bailey Process Automation, and is issued in strict confidence. Its use, or reproduction for use, for the reverse engineering, development or manufacture of hardware or software described herein is prohibited. No part of this document may be photocopied or reproduced without the prior written consent of Elsag Bailey.

Preface

@aGlance/IT is a multi-platform server/client product that is geared toward process control applications, using an industry accepted application programming interface (API). The @aGlance/IT provides a Dynamic Data Exchange/Object Linking and Embedding (DDE/OLE) interface for open access to Symphony Enterprise Management and Control System and INFI 90 OPEN Strategic Enterprise Management System. Therefore, customer specific client applications can be developed for multiple servers running on multiple platforms. For example, an @aGlance/IT client can use Microsoft Excel or Lotus 1-2-3 software to access information from an Elsag Bailey @aGlance/IT server without any API programming effort required.

The following consoles provide an @aGlance/IT server:

- OIS 40 series consoles.
- Signature 40 series consoles.
- Personal Work Center 90.
- Conductor VMS workstations.
- Conductor NT workstations.

This instruction provides general information on the @aGlance/IT servers and the following specific information for the OIS 40 series consoles, Signature 40 series consoles, and Conductor VMS workstations: how to obtain a license key (required for additional concurrent clients and servers), how to perform initial setup (network configuration), how to install an @aGlance/IT license key for additional servers and clients, and how to configure data access between multiple clients and servers. It also provides tag name attribute information, error messages, status messages and how to use the available diagnostic commands.

Revision B changes to this document include:

- Tag attributes updated to include Harmony I/O tag attributes.
- Inclusion of Conductor VMS and Conductor NT workstations.

List of Effective Pages

Total number of pages in this instruction is 101 consisting of the following:

Page No.	Change Date
Preface	Original
List of Effective Pages	Original
iii through vii	Original
1-1 through 1-7	Original
2-1 through 2-5	Original
3-1 through 3-6	Original
4-1 through 4-38	Original
5-1 through 5-10	Original
A-1 through A-25	Original
Index-1 through Index-3	Original

When an update is received, insert the latest changed pages and dispose of the superseded pages.

NOTE: On an update page, the changed text or table is indicated by a vertical bar in the outer margin of the page adjacent to the changed area. A changed figure is indicated by a vertical bar in the outer margin next to the figure caption. The date the update was prepared will appear beside the page number.

Table of Contents

	<i>Page</i>
SECTION 1 - INTRODUCTION.....	1-1
OVERVIEW	1-1
INTENDED USER.....	1-1
SYSTEM DESCRIPTION.....	1-2
SYSTEM REQUIREMENTS	1-2
FEATURES.....	1-3
INSTRUCTION CONTENT	1-4
HOW TO USE THIS INSTRUCTION	1-4
DOCUMENT CONVENTIONS	1-5
REFERENCE DOCUMENTS.....	1-5
NOMENCLATURE	1-6
GLOSSARY OF TERMS AND ABBREVIATIONS	1-7
SECTION 2 - DESCRIPTION AND OPERATION.....	2-1
INTRODUCTION.....	2-1
HARDWARE DESCRIPTION	2-1
SYSTEM DESCRIPTION.....	2-1
@aGlance/IT Servers.....	2-1
@aGlance/IT Client	2-2
@aGlance/IT SOFTWARE	2-2
Session Control.....	2-2
Supervisory Control	2-3
DATA EXCHANGE.....	2-3
Tagname Access.....	2-3
Real Time Data	2-4
Historical Data.....	2-4
Event Log Data	2-4
Monitoring.....	2-4
CLIENT ADD-INS	2-5
CONFIGURATION.....	2-5
SECTION 3 - INSTALLATION	3-1
INTRODUCTION.....	3-1
OVERVIEW	3-1
@aGLANCE/IT NETWORK CONFIGURATION.....	3-2
@aGLANCE/IT SERVER CONFIGURATION (WRITE ACCESS)	3-3
ADDING ADDITIONAL @aGLANCE/IT CLIENTS.....	3-4
@aGlance LICENSE	3-5
Registration	3-5
License Installation	3-5
SECTION 4 - SOFTWARE DETAILS.....	4-1
INTRODUCTION.....	4-1
SECURITY.....	4-1
UTI FUNCTION SUPPORT.....	4-1
@aGlance API FUNCTIONS	4-2
CLIENT LISTS	4-6
GET TAG ATTRIBUTE NAMES	4-7
SET OPTIONS	4-7
CALLING SECONDARY ATTRIBUTES	4-9
GET REALTIME TAG DATA.....	4-9

Table of Contents (continued)

	<i>Page</i>
<hr/>	
SECTION 4 - SOFTWARE DETAILS (continued)	
PUT REALTIME TAG DATA	4-10
MONITOR REALTIME TAG DATA	4-14
@aGlance/IT Server Performance	4-16
@aGlance/IT Server	4-16
GET HISTORICAL TAG DATA	4-17
AAG_GetHistory	4-19
AAG_GetHistEvent	4-20
AAG_GetHistStat	4-22
GET EVENT LOG DATA	4-24
GET WINDOW INFORMATION	4-30
REQUEST DISPLAY	4-31
WRITE TO DIAGNOSTIC LOG	4-32
ACTIVATE REMOTE TASK	4-33
Touch Point Activation	4-34
Key Select Activation	4-35
Function Key Activation	4-37
Command Processor Activation	4-37
<hr/>	
SECTION 5 - TROUBLESHOOTING	5-1
INTRODUCTION	5-1
SERVER DIAGNOSTICS	5-1
STATUS LIST MESSAGES	5-4
RETURN ERROR MESSAGES	5-7
<hr/>	
APPENDIX A - TAG ATTRIBUTE NAMES	A-1
INTRODUCTION	A-1
DATA TYPES	A-1
TAG ATTRIBUTE CHARACTERISTICS	A-1
TAG ATTRIBUTE NAMES	A-1
Primary Tag Attributes	A-2
Secondary Tag Attributes	A-15
<hr/>	

List of Figures

<i>No.</i>	<i>Title</i>	<i>Page</i>
1-1.	@aGlance/IT Application Overview	1-3
2-1.	@aGlance/IT Server/Client Architecture	2-2
4-1.	Get Events Example (Time Span Retrieval)	4-29
4-2.	GetEvents Example (Reference Time Retrieval)	4-29

List of Tables

<i>No.</i>	<i>Title</i>	<i>Page</i>
1-1.	Server Platforms	1-2
1-2.	Reference Documents	1-5
1-3.	Nomenclature (ODS)	1-6
1-4.	Nomenclature (ODC)	1-6
1-5.	Glossary	1-7
4-1.	UTI/@aGlance Functions	4-1
4-2.	Supported @Glance/IT API Functions	4-5
4-3.	Set Options Arguments	4-7
4-4.	Writable Attributes, Console Level	4-10
4-5.	Writable Attributes, Module Level	4-12
4-6.	Trend Types for Historical Attributes	4-18
4-7.	Time Span Retrieval Parameters	4-25
4-8.	Returnable Attributes	4-26
4-9.	Reference Time Retrieval Parameters	4-28
4-10.	Input Value (GetWindow)	4-30
4-11.	Output Values (GetWindow)	4-30
4-12.	Input Values (RequestDisplay).....	4-32
4-13.	Input Value (DiagnosticLog)	4-32
4-14.	Input Values (User Task Activation)	4-33
4-15.	Command Parameters (Touch Point).....	4-35
4-16.	Command Parameters (Key Select)	4-36
4-17.	Screen Fields (User Task Definition)	4-37
4-18.	AAG_CP Commands	4-38
4-19.	AAG_CP Command Specification File.....	4-38
5-1.	Diagnose Command Options	5-2
5-2.	Status Messages	5-4
5-3.	Error Messages	5-7
A-1.	Common Attributes, All Tag Types.....	A-2
A-2.	ANALOG Tag Attributes.....	A-3
A-3.	DAANALG Tag Attributes	A-4
A-4.	DADIG Tag Attributes	A-5
A-5.	DAANG (DANG Tag Type) Tag Attributes	A-6
A-6.	DADIGTL/INTDIG Tag Attributes	A-7
A-7.	Device Driver (DD) Tag Attributes	A-7
A-8.	Digital Tag Attributes	A-8
A-9.	INTANG Tag Attributes	A-9
A-10.	Multistate Device Driver (MSDD) Tag Attributes	A-9
A-11.	N90STA Tag Attributes.....	A-10
A-12.	Remote Control Memory (RCM) Tag Attributes	A-11
A-13.	RMCB Tag Attributes	A-11
A-14.	RMSC Tag Attributes.....	A-12
A-15.	Station Tag Attributes	A-13
A-16.	TEXT Tag Attributes.....	A-14
A-17.	TEXTSTR Tag Attributes	A-14
A-18.	All Tag Type Attributes (Secondary)	A-15
A-19.	Analog/INTANG Tag Attributes (Secondary).....	A-16
A-20.	DAANALG Attributes (Secondary)	A-17
A-21.	DAANG Attributes (Secondary)	A-18
A-22.	DADIG Tag Attributes (Secondary).....	A-20
A-23.	Device Driver (DD) Tag Attributes (Secondary)	A-21
A-24.	DEVSTAT Tag Attributes (Secondary)	A-21
A-25.	Digital/DADIGTL/INTDIG Tag Attributes	A-21

List of Tables (continued)

<i>No.</i>	<i>Title</i>	<i>Page</i>
A-26.	MSDD Tag Attributes (Secondary)	A-22
A-27.	N90STA Tag Attributes (Secondary).....	A-22
A-28.	RCM Tag Attributes (Secondary)	A-23
A-29.	RMCB Tag Attributes (Secondary)	A-23
A-30.	RMSC Attributes (Secondary).....	A-24
A-31.	Station Tag Attributes (Secondary).....	A-24
A-32.	TEXT Tag Attributes (Secondary)	A-25
A-33.	TEXTSTR Tag Attributes (Secondary)	A-25

Trademarks and Registrations

Registrations and trademarks used in this document include:

™ @aGlance/IT	Trademark of Intuitive Technologies Corporation.
™ AXP	Trademark of Digital Equipment Corporation.
™ Motif	Trademark of Open Software Foundation, Incorporated.
™ Open VMS	Trademark of Digital Equipment Corporation.
™ VAX	Trademark of Digital Equipment Corporation.
™ VMS	Trademark of Digital Equipment Corporation.
™ Windows NT	Trademark of Microsoft Corporation.
® HP-UX	Registered trademark of Hewlett-Packard Company.
® INFI 90	Registered trademark of Elsasg Bailey Process Automation.
® Lotus	Registered trademark of Lotus Development Corporation.
® Microsoft	Registered trademark of Microsoft Corporation.
® MS	Registered trademark of Microsoft Corporation.
® SCO	Registered trademark of The Santa Cruz Operation, Incorporated.
™ Symphony	Trademark of Elsasg Bailey Process Automation.
® UNIX	Registered trademark of UNIX System Laboratories.
® Windows	Registered trademark of Microsoft Corporation.
® Windows NT	Registered trademark of Microsoft Corporation.

SECTION 1 - INTRODUCTION

OVERVIEW

@aGlance/IT is a multi-platform server/client product that is geared toward process control applications, using an industry accepted application programming interface (API). The @aGlance/IT provides a Dynamic Data Exchange/Object Linking and Embedding (DDE/OLE) interface for open access to Symphony Enterprise Management and Control System and INFI 90 OPEN Strategic Enterprise Management System. Therefore, customer specific client applications can be developed for multiple servers running on multiple platforms. For example, an @aGlance/IT client can use Microsoft Excel or Lotus 1-2-3 software to access information from an Elsag Bailey @aGlance/IT server without any API programming effort required.

An Elsag Bailey @aGlance/IT server provides the client application with information and services such as session control, data acquisition, real time and historical data, tag data and attributes, and other miscellaneous functions. This imported data can be then used in available Elsag Bailey tools and various third party tools such as relational databases, expert systems, simulation networks, report generators, statistical analysis packages, scheduling and planning packages, tools, charting tools, and graphic tools.

INTENDED USER

This document is an @aGlance/IT server/client installation and setup manual. It is intended to be used by personnel using the @aGlance/IT server/client including system operators, engineers, programmers, plant managers and supervisors, and corporate management and personnel (such as MIS personnel).

For installation, system set-up and configuration, engineers and programmers should have a technical knowledge of the server/client platform being used. Knowledge of the applicable communication system and the third party application software being implemented is also required. Experience installing and configuring Symphony and INFI 90 OPEN systems and applications is beneficial.

System operators, plant managers and supervisors, and corporate personnel should have a working knowledge of server/client operation on the applicable platform. Use of the third party application software is beneficial. Experience operating Symphony and Symphony and INFI 90 OPEN systems and applications is beneficial.

SYSTEM DESCRIPTION

The @aGlance/IT server/client product is layered on various Elsag Bailey platforms. It provides an open interface between a server residing on the Elsag Bailey system and client platforms running third party applications such as Excel or Lotus 1-2-3 spreadsheets. The @aGlance/IT API takes care of the network layer on both the client and server side of the interface.

The @aGlance/IT server is shipped with the platforms listed in Table 1-1. Figure 1-1 shows an overview of the various platforms that can be accessed using @aGlance/IT software.

NOTES:

1. Any reference to the OIS 40 Series consoles also applies to the Signature 40 series consoles.
2. The OIS 40 Series and Conductor VMS @aGlance/IT server operate identically and, with the exception of Sections 1 and 2, the information in this document specifically applies to the OIS 40 Series and Conductor VMS platforms.
3. For additional information on the PWC 90 and Conductor NT platforms, refer to the appropriate instruction listed in Table 1-2.

Table 1-1. Server Platforms

Server Platform	Operating Systems
Conductor NT	Windows NT
Conductor VMS	Alpha AXP OpenVMS
Operating Interface Station 40 Series Release H.2 or greater	Alpha AXP OpenVMS
PWC 90	Windows NT
Signature 40 Series Release H.2 or greater	Alpha AXP OpenVMS

SYSTEM REQUIREMENTS

Any @aGlance/IT client can connect to any @aGlance/IT server provided the minimum network requirements are met to access the specific @aGlance/IT server.

OIS 40 Series and Conductor VMS

The requirements to use an @aGlance/IT server on a console or workstation are:

- The node on which the @aGlance/IT server resides must be connected to the same TCP/IP communication network on which the @aGlance/IT clients reside.

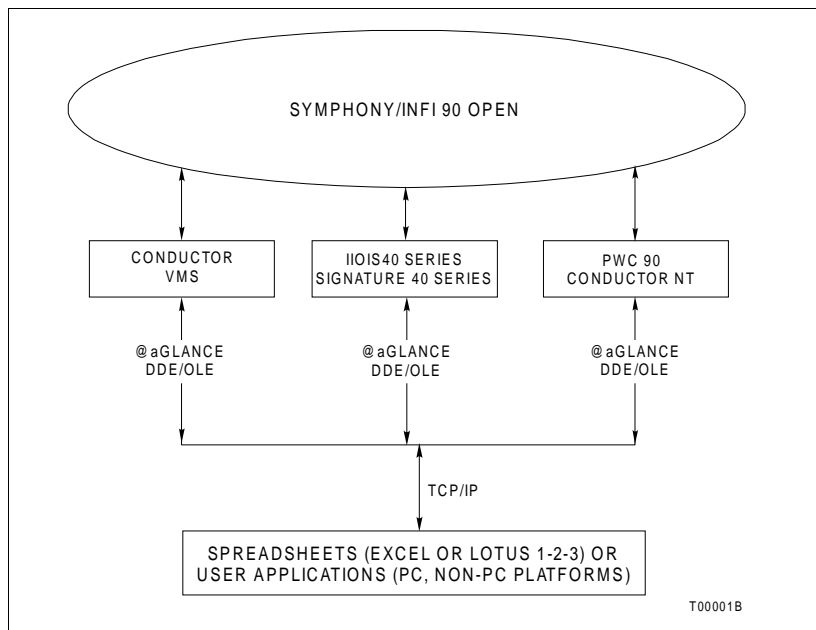


Figure 1-1. @aGlance/IT Application Overview

- For additional users (multiple clients and servers) an @aGlance/IT server license key must be properly installed. When @aGlance/IT software is shipped with a console or workstation, it is shipped with one server and support for one client without a license key.

**PWC 90 and
Conductor NT**

Refer to the appropriate configuration instruction listed in Table 1-2.

FEATURES

- Enables multiple client platforms to run client applications with commonly used spreadsheet applications in the Windows environment such as Excel and Lotus1-2-3.
- Provides real-time, historical and event data gathering for report writing, system analysis, system performance records, etc.
- Interfaces multiple client platforms to the @aGlance/IT servers without additional hardware required, the @aGlance/IT API provides the functional interface.
- Affords versatility in the data collecting effort, every licensed client can be configured for unique data collection duties.
- Harmony I/O attributes are available on OIS 40 Series consoles with release J.1 software or later and Conductor VMS workstation version 1.0.

INSTRUCTION CONTENT

This instruction contains software setup information for @aGlance/IT servers and clients. This instruction contains the following sections:

Introduction	Provides an overview of the @aGlance/IT server/client software. Other items include intended user information, features, a list of reference documents (Table 1-2), software nomenclature (Table 1-3), and glossary of terms and abbreviations (Table 1-5).
Description and Operation	Contains a functional description of the Open Data Server/Client.
Installation	Outlines how to initially configure the @aGlance/IT server, network configuration, add @aGlance/IT clients, how to install a license key on consoles or workstations when adding servers, and how to setup additional clients for those servers.
Software Details	This section details supported @aGlance/IT functions, system configuration, UTI function support, how to generate client lists, server setup, writable attributes, and configuration of real-time and historical data monitoring and event monitoring.
Troubleshooting	This section lists troubleshooting information which includes use to the Diagnose command, status messages, and error messages.
Tag Attribute Names	This appendix contains a complete list of tag attribute names and includes a description of each attribute, data type, and valid attribute values.

HOW TO USE THIS INSTRUCTION

1. Read the introduction and the description and operation sections. Reading these sections helps the user understand the @aGlance/IT server software and how it fits in with the Symphony and INFI 90 OPEN systems and third party application software.
2. Follow the installation steps in **Section 3** to perform installation for OIS 40 Series and Conductor VMS platforms.
3. Read **Section 4** for detailed information on the configuration and use of the @aGlance/IT software on OIS 40 Series and Conductor VMS platforms.
4. Refer to **Section 5** for diagnostic and troubleshooting information for OIS 40 Series and Conductor VMS platforms.
5. Refer to the **Appendix A** for tag attribute information for OIS 40 Series and Conductor VMS platforms.

DOCUMENT CONVENTIONS

This document uses standard text conventions throughout to represent keys, user data input and display items.

KEY Identifies a keyboard key.

Example: Press **Return**.

Display item Any item that displays on the screen appears as italic text in this document.

Example: *No Data*

File name Any file names and file extensions appear as bold-italic text.

Example: ***Program.exe***

BOLD Indicates user input that must be entered exactly as shown.

Example: **set file/prot=w:re file.dat**

Italic Identifies a variable parameter in a command line.

Example: *nodename hostid (port-number)*

[] A user input that is optional. Text within the bracket still follows the previously described conventions.

Example **command, [start index], [stop index]**

REFERENCE DOCUMENTS

Table 1-2 contains a list of documents related to the Open Data Server and referenced in this instruction.

Table 1-2. Reference Documents

Number	Title
I-E-96-192-2	Configuration (IS42 and IIOIS42)
I-E96-197	Display Builder Reference (IS42 and IIOIS42)
WBPEEUI220023B0	File Utilities (IS42 and IIOIS42)
WBPEEUI220756A0	OIS40 Series Operation
WBPEEUI220757A0	OIS40 Series Configuration
WBPEEUI220758A0	OIS40 Series File Utilities
WBPEEUI220759A0	OIS40 Series Display Builder
WBPEEUI220762A0	Configuration, Personal Work Center 90
WBPEEUI220773A0	Operation, Conductor VMS
WBPEEUI220774A0	Configuration, Conductor VMS
WBPEEUI220775A0	File Utilities, Conductor VMS

Table 1-2. Reference Documents (continued)

Number	Title
WBPEEU1220776A0	Display Builder Reference
WBPEEU1220779A0	Configuration, Conductor NT
WBPCEU1900014A0	@aGlance/IT User's Guide consisting of: @aGlance/IT for Microsoft Windows @aGlance/IT for Microsoft Excel @aGlance/IT for Lotus 1-2-3
WBPCEU1900020A0	@aGlance/IT Installation and Operation

NOMENCLATURE

Table 1-3 lists Open Data Server nomenclature. Table 1-4 lists Open Data Client nomenclature.

Table 1-3. Nomenclature (ODS)

position	1	2	3	4	5	6	7	8	9	
	L	I	O	D	S	?	?	?	?	Open Data Server
						1				Class @aGlance/IT
							1			Platform OIS 40 series, Signature 40 series and Conductor VMS
								1		Simultaneous Clients 5 (1 server) 10 (2 servers)
								2		
									0	Reserved

Table 1-4. Nomenclature (ODC)

position	1	2	3	4	5	6	7	8	9	
	L	I	O	D	C	?	?	?	?	Open Data Client
						1				Class @aGlance/IT
							1			Operating System MS Windows (3.1, 3.11, 95, NT)
								1		Media 3.5 in., 1.44 Mbyte floppy
									0	Reserved

GLOSSARY OF TERMS AND ABBREVIATIONS

Table 1-5 contains definitions of terms and abbreviations that are unique to Elsasg Bailey.

Table 1-5. Glossary

Term	Definition
Establish	A mechanism which enables exception reports to flow between system nodes.
Node	A point of interconnection to a network.
Node Address	A unique identifier of a specific device or a communication channel. Refers to Plant Loop, Superloop or INFI-NET address.
OIS	Operator interface station. Integrated operator console with data acquisition and reporting capabilities. It provides a digital access into the process for flexible control and monitoring.

SECTION 2 - DESCRIPTION AND OPERATION

INTRODUCTION

This section contains a brief description of the @aGlance/IT server/client. @aGlance/IT is a multi-platform server/client geared towards process control applications using an industry standard application programming interface (API). The @aGlance/IT API provides open access to Symphony and INFI 90 OPEN data which enables user to develop @aGlance/IT server/client application using commercially available software tools such as spreadsheets, relational databases, expert systems, simulation networks, statistical analysis packages, and report generation tools.

HARDWARE DESCRIPTION

The @aGlance/IT data server is available on the following products:

- Operator Interface Station 40 and Signature 40 Series release H.2 or greater.
- PWC 90.
- Conductor VMS.
- Conductor NT.

Client platforms supported by the @aGlance/IT API include:

- Windows 3.1.
- Windows 95.
- Windows NT.

NOTE: Other client platforms may be supported as they are developed and released by Intuitive Technology Incorporated.

Server to client communication requires a TCP/IP network and the appropriate TCP/IP software for the client platform.

SYSTEM DESCRIPTION

The @aGlance/IT server/client API is capable of simultaneous operation on any mix of multiple server/client platforms. For example, on a system having Windows 95 and Windows NT clients, the clients can simultaneously access data from any @aGlance/IT server platform within the system.

@aGlance/IT Servers

The @aGlance/IT server provides open access to process control system data using a Dynamic Data Exchange/Object

Linking and Embedding (DDE/OLE) interface in a server/client environment. This open data access to the process control system is made possible by mapping the @aGlance/IT API into the existing Elsag Bailey API. Consequently, client applications running on different platforms can simultaneously communicate with the @aGlance/IT server. The @aGlance/IT server provides client applications to system services such as session control, data acquisition, real time and historical data, tag data and attributes, and other miscellaneous functions.

@aGlance/IT Client

An @aGlance/IT client application incorporates the @aGlance/IT client function library, which provides callable routines for initialization and for the @aGlance/IT services. The spreadsheet add-ins have incorporated calls to the @aGlance/IT client function library. A typical client calls the @aGlance/IT initialization routine and then continues to perform its normal functions. Figure 2-1 shows the elements of the @aGlance/IT server/client components.

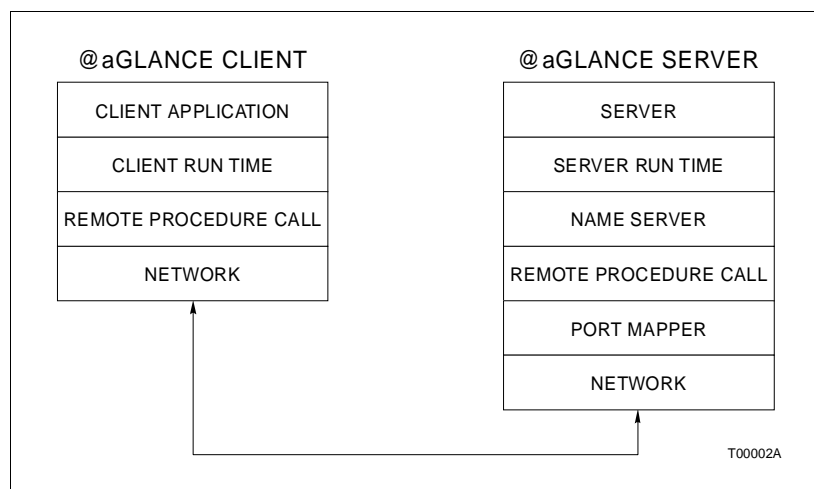


Figure 2-1. @aGlance/IT Server/Client Architecture

@aGlance/IT SOFTWARE

All @aGlance/IT servers support following software security functions:

- Session control.
- Client connections.
- Supervisory control.

Session Control

The @aGlance/IT server supports a limited number of concurrent client connections. If the server receives a connection

request that is greater than the maximum number of connections allowed, that connection request is denied and an error message is sent to the log file.

A client application must initiate a connection request using the **AAG_ClientInit** function. This function call registers a client with the @aGlance/IT data server. The client application must also end a session using **AAG_ClientTerminate** function. A client must end the session with this function call or the session remains dangling.

Supervisory Control

The @aGlance/IT data servers can restrict connections to authorized users and also limit write access to authorized users. There are no access restrictions to individual tags. The standard security mechanisms available through @aGlance/IT are proxies and permissions.

Proxies determine which clients can connect to a server. A specific user on a specific node can be given access, all users on a specific node can be given access, or all users on all nodes can be given access. Permissions determine which server functions that a client can perform.

Permission names are server specific and are not defined by the @aGlance/IT API. Each server has permission names assigned so that there is no conflict with permissions on other @aGlance/IT servers. The @aGlance/IT API provides a utility for configuring proxies and permissions. Permissions can be checked by the server by issuing an **AAG_CheckPermission** function which passes the permission name and returns if the permission is granted or not.

DATA EXCHANGE

The @aGlance/IT servers are capable of providing five types of data exchange. Those data exchange types are:

- Tagname access.
- Real time data.
- Historical data.
- Event log data.
- Monitoring.

Tagname Access

@aGlance/IT client access to an @aGlance/IT server database is based on use of consistent tagnames. Client applications can access the @aGlance/IT database, and perform data read and write operations using these tagnames. Tagname based searches are supported for wild card searches (i.e., using the

asterisk character in the tagname field) and single character searches (i.e., using the question mark in the tagname field).

Real Time Data

@aGlance/IT servers support both read and write access to real time data. Write access requires supervisory control privileges. If a client application does not have supervisory control privileges, it cannot write data back to the data server or to the control system.

Historical Data

Historical data read is supported by all @aGlance/IT servers. Historical data write capability is supported by all @aGlance/IT servers except the OIS 40 Series and Conductor VMS @aGlance servers. Write access requires supervisory control privileges. If a client application does not have supervisory control privileges, it cannot write data to the control system or server. The @aGlance/IT provides both interpolated and compressed data for historical data requests.

Event Log Data

OIS 40 Series and Conductor VMS servers support event log data retrieval using the **AAG_Command** function. This includes alarms, events, and operator actions. The server specific command to get events log data is **GetEvents**. Event log data is retrieved for a time period by supplying a start and end time. It can also be retrieved by supplying a start time and direction of search, forward or backward.

The **GetEvents** command is used to retrieve all events saved to the hard disk. It cannot retrieve any event log data archived on an optical disk or any other medium. Refer to [Section 4](#) for more detailed information on the **GetEvents** command and an example of how it is used.

Monitoring

@aGlance/IT servers support real time data monitoring. This gives client applications the capability of reading a set of data points at time intervals or when a significant value change occurs for any of the data points. Client applications have the capability of taking points in and out of monitoring. The data server drops the monitor points if a client application terminates without notification. However it is recommended that the client application formally terminate any active monitor requests before terminating itself.

CLIENT ADD-INS

An @aGlance/IT client can be provided as an add-in to commonly used spreadsheet applications such as Excel and Lotus 1-2-3. The add-ins allow the spreadsheet users to interactively request data from the @aGlance/IT servers. Data functions that are not part of the @aGlance/IT API and are implemented as part of the @aGlance/IT server use the **AAG_Command** function.

CONFIGURATION

@aGlance/IT servers provide server configuration to @aGlance/IT clients. The server provides a list of tagnames and a list of attributes for any specified tag. The server supports both wild card search and single character search in the tagname field.

SECTION 3 - INSTALLATION

INTRODUCTION

This section contains an overview of @aGlance/IT software installation and licensing procedures. Installation steps include @aGlance/IT network configuration, how to assign client permissions for write access to the server, how to add @aGlance/IT clients to a network, how to register for and receive a software license key to permit activation of multiple @aGlance/IT server/clients, and the steps required to activate a multiple server/client license key. Read the installation procedure descriptions outlined in **OVERVIEW** to determine which procedures are required for a particular system.

OVERVIEW

The @aGlance/IT server software is licensed to prevent the use of unauthorized copies of the data server. Refer to Tables 1-3 and 1-4 for nomenclature which explains the available @aGlance/IT server/client configurations.

The Elsag Bailey @aGlance/IT server is shipped with one unlicensed server capable of serving one client. A license key is required to run more than one server or to serve more than one client concurrently. One data server can support five connected clients simultaneously. One or more copies of the server can be licensed and running on a given machine.

The installation procedures in this section cover:

- **@aGLANCE/IT NETWORK CONFIGURATION.** This procedure explains how to establish network connections for servers and clients. Complete the steps in this procedure to configure the network for the one client @aGlance/IT server that is shipped with the platform software. This procedure is also required when adding licensed servers capable of serving simultaneous clients.

NOTE: This procedure is also found in the appropriate *File Utilities* manual. It does not need to be repeated if already performed.

- **@aGLANCE/IT SERVER CONFIGURATION (WRITE ACCESS).** These instructions explain on how to assign permissions to clients for write access to a server. Complete the steps in this section to assign the **EBOIS-WRITE** permission to clients.
- **ADDING ADDITIONAL @aGLANCE/IT CLIENTS.** Additional clients can be added to the system after the network

is configured. Use this procedure to add clients to the system.

- **@aGlance LICENSE.** This two part procedure covers @aGlance/IT license registration and installation. Complete these procedures to register and install a software license for additional servers that are capable of servicing concurrent clients.

After the software license is installed, perform the required network configuration for the added server, assign client write permissions as required, and add additional clients as required to meet the system needs. Refer to the appropriate procedure as necessary.

To install client software, refer to the appropriate @aGlance/IT installation documentation. Table 1-2 lists document numbers.

@aGLANCE/IT NETWORK CONFIGURATION

@aGlance/IT requires that network connections for clients and servers be established. These network connections are established using the @aGlance/IT Administration utility program.

NOTE: Before the network connection(s) for the @aGlance/IT can be established the console has to be configured on the TCP/IP network with the PORTMAPPER feature enabled and the TCP/IP services started. Refer to the appropriate *File Utilities* manual listed in Table 1-2 for more information.

To establish @aGlance/IT client/server network connections:

1. Open a terminal window and log into the SYSTEM account.
2. To verify that the network connection for has been established, type:

```
UCX PING console_name 
```

where:

console_name Host name of the local console.

3. If not already defined, add the host name and address for any remote client/server that will access the local console. To add a remote client/server, type the following at the \$ prompt:

```
ADDHOST client_name aa.bb.cc.dd 
```

where:

client_name @aGlance/IT client name.

aa.bb.cc.dd @aGlance/IT client name.

NOTES:

1. This command also updates the IIOIC42 host database configuration file. Refer to the appropriate **File Utilities** manual listed in Table 1-2 for more information.
2. There is no need to re-enter host names of remote consoles that were previously entered when configuring the TCP/IP Network. Refer to the appropriate **File Utilities** manual listed in Table 1-2 for more information.
4. To display a list of all client/servers along with all previously added remote consoles, type:

SHOWHOSTS

5. Define local console node in the local @aGlance/IT server host database. This allows the @aGlance/IT Administration utility program to list the servers running on this network node.

To define the local console network node, type the following at the \$ prompt:

RUN AAG\$SYSTEM:AAG_ADMIN

The @aGlance/IT Administration menu will appear. Refer to the *@aGlance/IT System Manual* for information on how to use this utility.

6. Reboot the console. Type the following at the \$prompt:

REBOOT

After rebooting, the console will have a single @aGlance/IT server with the default name of EBOIS. This server will have the ability to provide access to one @aGlance/IT client at a time.

@aGLANCE/IT SERVER CONFIGURATION (WRITE ACCESS)

The @aGlance/IT feature provides the capability for @aGlance/IT clients to access configured @aGlance/IT servers on the network, so that the client can acquire real-time, historic, and event log data.

The console or workstation is initially configured with limited security restrictions for all @aGlance/IT clients. All clients are initially assigned to the **SYSTEM** account and are provided the following permissions: **AAG_ListServers**, **AAG_StopSession**, and **AAG_StopServer**. These permissions allow all defined clients to perform read operations as well as the ability to list the @aGlance/IT servers, stop a client session, and stop a server.

An additional permission, that provides the client the capability to write to a server, is also available. The **EBOIS-WRITE** permission can be granted to any client. The **AAG_Admin** utility is used to change permission settings.

To assign an **EBOIS-WRITE** permission to a client:

1. Log in using the **SYSTEM** account.
2. At the \$ prompt type:


```
run aag$system:aag_admin
```
3. Select the *Server Setup* menu to perform client access configuration. Follow the menu through to *Client Permissions*. The **@aGlance/IT Installation and Operation** manual shows details of these menus.
4. Add the permission **EBOIS-WRITE** to the list of local client permission types.
5. Permission assignments do not take effect until the @aGlance/IT server restarts. Exit the system account and reset the system. Refer to the appropriate configuration manual listed in Table 1-2 for system reset procedures.

Refer to the **@aGlance/IT Installation and Operations** manual for a discussion on proxy accounts, permission assignments and the use of the **AAG_Admin** utility.

ADDING ADDITIONAL @aGLANCE/IT CLIENTS

Additional @aGlance/IT clients can be added after the network has been configured. To add an additional @aGlance/IT client:

1. On the local console, open a terminal window, logging into the **SYSTEM** account.
2. To add the additional client, type the following at the \$ prompt:

```
ADDHOST client_name aa.bb.cc.dd Return
```

where:

client_name @aGlance/IT client name.

aa.bb.cc.dd @aGlance/IT client address.

3. To verify that the network connection for the client has been established, type:

UCX PING *client_name*

@aGlance LICENSE

To obtain a software license (license key) that permits installing additional @aGlance server/clients, follow the steps listed under **Registration**. To install additional server/clients using a valid license key, follow the steps under **License Installation**.

Registration

The platform software is released with a single server capable of servicing one client at a time. Licensed servers can be added that are capable of servicing up to five clients simultaneously. These added servers require a software license key before they can be installed.

The required software license key is generated by registering the @aGlance/IT software with Elsas Bailey. There are two methods available for software registration: phone registration through Elsas Bailey Customer Services Center or registration via a computer with a modem through the Elsas Bailey Process Automation License Administration System. The following items are needed to register the @aGlance/IT software.

- The *HOSTID* which is generated using the **Imhostid** command. Refer to Step 1 under **License Installation**.
- The Elsas Bailey Software Registration Form which contains license registration instructions.

License Installation

Complete the following steps to install additional @aGlance/IT server/clients having a valid software license key.

1. Obtain the licensing *hostid* for the desired machine. Log in to the SYSTEM account and execute the **Imhostid** command.

2. Register the @aGlance/IT software by following the instructions on the Elsag Bailey Software Registration Form. Elsag Bailey will return a license key and checksum to be inserted into the **flexlm** upgrade license procedure.
3. Log in with the system account to the console or workstation to be upgraded.
4. At the \$ prompt, type:

```
setaaglicense 
```

5. Enter the appropriate information at the prompt and press :

```
NODENAME FOR THIS NODE
```

Enter the *NODENAME* which is the TCP/IP hostname assigned to this console.

```
HOSTID (From LMHOSTID utility)
```

Enter the *HOSTID* that was obtained in Step 1.

```
NUMBER OF SERVERS LICENSED
```

Enter the number of server licenses obtained in Step 2.

```
LICENSE KEY (From Elsag Bailey)
```

Enter the license key returned from Elsag Bailey during the license registration process in Step 2.

```
CHECKSUM (From Elsag Bailey)
```

Enter the checksum returned from Elsag Bailey during the license registration process in Step 2.

6. After the required information is entered, the command procedure takes the input and generates a license file (**flexlm.dat**) in this directory. It verifies the license file by executing a checksum utility and displaying the result at the console. If the checksum utility indicates a valid license file type **Y** at the prompt. If the checksum utility indicates an invalid license file, type **N** to abort and enter the data again.

If the checksum utility fails, verify the *HOSTID* entered is correct and that the correct *HOSTID* was supplied to Elsag Bailey during the software registration. Verify that the number of servers licensed was entered correctly. Verify the license key and checksum returned by Elsag Bailey were correctly entered.

7. If the checksum utility verifies the license file, re-boot the machine to activate the new license file.

SECTION 4 - SOFTWARE DETAILS

INTRODUCTION

This section provides a brief description of supported @aGlance/IT functions. Refer to the **@aGlance/IT System User's Guide** for details on each function. Also covered in this section are system configuration details which include: UTI function support, how to generate client lists, server setup using the **SetOptions** routine, a list of console and module level writable attributes, and how to configure the server for real-time and historical data monitoring, and event monitoring.

SECURITY

The @aGlance/IT server maintains system integrity and checks on the maximum number of client connections supported. The **AAG_ClientInit** function registers a client with an @aGlance/IT server. The spreadsheet add-in issues this command automatically for the user.

UTI FUNCTION SUPPORT

The OIS40 Series console and Conductor VMS workstation support functionality supplied by the User Task Interface (UTI) functions and @aGlance/IT functions listed in Table 4-1. All UTI function do not map to a unique command in the @aGlance server. Some UTI functions are combined to form a command in the server using the **AAG_Command** function. For complete details on UTI functions refer to the **User Task Interface** instruction. This section lists the supported @aGlance API functions with a brief description, refer to the **@aGlance/IT User's Manual** for complete details.

Table 4-1. UTI/@aGlance Functions

UTI Function	@aGlance API Function
uti_get_crt_info	AAG_Command
uti_create_xdspnam	Not supported
uti_destroy_xdspnam	Not supported
uti_request_display	AAG_Command
uti_diagnostic_log	AAG_Command
uti_strerror	AAG_GetErrorText
uti_print_text_file	Not supported
uti_define_reports	Not supported
uti_fcp_mode	Not supported
uti_output_values	Not supported

Table 4-1. UTI/@aGlance Functions (continued)

UTI Function	@aGlance API Function
uti_connect	AAG_StartSession
uti_disconnect	AAG_EndSession
uti_get_tag_indices	Not supported
uti_output_stn_value	AAG_PutList, AAG_PutTable
uti_set_stn-values	AAG_PutList, AAG_PutTable
uti_read_values	AAG_GetList, AAG_GetTable
uti_write_int_values	AAG_PutList, AAG_PutTable
uti_read_tag_data	AAG_GetList
uti_get_ois_time	Not supported
uti_set_oas_time	Not supported
uti_convert_time	Not supported
uti_get_trend	AAG_GetHistory
uti_get_parameters	AAG_Command
Event Log Data	AAG_Command

The following commands and requests are implemented through the **AAG_Command**.

- **Get Event Log Data.**
- **Get Window Information.**
- **List Clients.**
- **Request Display.**
- **Diagnose.**
- **Diagnostic Log.**
- **Activate Remote Tasks.**
- **Set Options.**

@aGlance API FUNCTIONS

The following @aGlance/IT API functions are supported by OIS 40 Series consoles and Conductor VMS workstations. Refer to Table 4-2 for a complete list of @aGlance/IT functions supported. For more information on these API functions refer to the **@aGlance/IT User's Guide**.

AAG_CancelMonitor	This function cancels a specific monitor request. This call can be made either by a client or a server.
AAG_ClientInit	This function initializes the client @aGlance/IT library. A client invokes this function before using any other @aGlance/IT functions. There is no need to call this function for spreadsheet add-in and Visual Basic applications.
AAG_ClientTerminate	This function terminates all sessions between the client and @aGlance/IT servers. This call must be issued by a client to prevent any dangling connections. There is no need to call this function for spreadsheet add-in and Visual Basic applications.

AAG_Command	This function provides a mechanism for invoking server specific functions that do not fit other @aGlance/IT functions. The client specifies a string that represents a command to the server.
AAG_EndSession	This function ends a client session on the specified server and is used by clients only.
AAG_GetAttr	This function returns attribute names for a set of tags which belong to the specified server.
AAG_GetErrorText	This function returns a pointer to a character string containing an @aGlance/IT error message.
AAG_GetHistEvent	This function allows a client to retrieve compressed historical data from a server. The client specifies a start and an end time, and the server returns historical values that are recorded by the data historian (i.e., compressed and without interpolation). The start time and the end time are text string fields, and cannot be numeric fields and cannot be EXCEL time or data values. The required format of the start time and the end time is <i>dd-mmm-yyyy hh:mm:ss</i> .
AAG_GetHistory	This function returns the historical values for a given list of tag and attribute names. The client specifies the start time, the interval time, and number of time intervals. If no data is found for a data period, this function returns interpolated data. The start time and the interval time are text string fields, they cannot be numeric fields and cannot be EXCEL time or data values. The required format of the start time is <i>dd-mmm-yyyy hh:mm:ss</i> and the interval time is <i>ss</i> (integer value of seconds, 32,000 seconds maximum).
AAG_GetHistStat	This function allows a client to retrieve statistical data from a server. Statistical values such as the maximum, the minimum, and the average over a time period are returned. The client application provides input such as the tag list, tag count, start time, end time, type list, and value table. The start time and the end time are text string fields, they cannot be numeric fields and cannot be EXCEL time or data values. The required format of the start time and the end time is <i>dd-mmm-yyyy hh:mm:ss</i> . The type list is a list of text strings containing the statistical types. The recommend standard types as they are defined in the header file AAGDEF.H are: AAG_HIST_MIN minimum AAG_HIST_MAX maximum AAG_HIST_AVG average
AAG_GetList	This function fetches the values for the specified list of tag and attribute pairs. It is supported by all Elsag Bailey @aGlance/IT servers.

- AAG_GetTable** This function obtains the values for the specified table of tags and attributes from the specified server. For *m* tags and *n* attributes, *m* by *n* values are returned. It is supported by all Eltag Bailey @aGlance/IT servers.
- AAG_GetTags** This function returns tag names known to the indicated server. It is supported by all Eltag Bailey @aGlance/IT servers.
- AAG_Monitor** This function requests the server to monitor a specified data point for changes in value. Initial data point values are returned immediately. Updated values and status information are made available to the client through the **Monitor_Update** callback. The client provides information such as tag list, attribute list, list count, value list, and other optional parameters such as status list, monitor type, interval, monitor arguments, argument count, client data, and monitor ID. The optional parameter monitor type indicates the type of monitoring to be performed. If used, it must be one of the following: text strings, “**AAG_POLLED**” or “**AAG_EXCEPTION**”. The optional parameter interval indicates how often the server should check the monitored data points for changes (for polled type monitoring). If used, must contain time data as text string, it cannot be a numeric field or an EXCEL time or data value. The required time interval format *ss* (integer) represents the number of seconds at which the data is to be polled. If the monitor type is not specified in the **AAG_Monitor** command, then the server defaults the monitor type to the mode specified for the client (refer to **SET OPTIONS**).
- AAG_MonitorInit** This function initializes the client for monitoring activity. The client uses this function after calling **AAG_ClientInt** function and before making any monitor request with **AAG_Monitor** or dispatching events with **AAG_Dispatch**.
- AAG_MonitorTerminate** This function performs clean up after a client has finished all monitoring activity. This function is optional and if used, must be called before the **AAG_ClientTerminate** function.
- AAG_PutHistory** This function enables the client to write data into a server historical database. The client provides information such as tag list, attribute list (optional), list count, time list (optional), value list, status list (optional), and put type (optional). The time list, if provided must contain time data as text string field and it cannot be a numeric field or an EXCEL time or data value. The required format of time values in the time list is *dd-mm-yyyy hh:mm:ss*. The optional parameter put type must also contain test string as data and the recommended put type is **AAG_INSERT** and **AAG_REPLACE**. This function is not supported by the OIS 40 Series or Conductor VMS @aGlance/IT servers.
- AAG_PutList** This function writes the values specified into the data points specified by the tag attribute pairs. It is used by client

applications to write real time data. A client application requires the **EBOIS_WRITE** permission to use this function.

AAG_PutTable	This function writes the values for the specified table of tags and attributes. For m tags and n attributes, m by n values are provided. This function is used by client applications to write real time data. A client application requires the EBOIS_WRITE permission to use this function.
AAG_Shutdown	This function sends a request to the server that the server shut itself down. The server may refuse this request depending on the client identity and its permissions, or on other client sessions in progress. Some servers that have not registered the corresponding callback function will always refuse this request. A client application requires the AAG_StopServer permission to use this function.
AAG_StartSession	This function may be used by a client to establish a session on a server before any data access is performed. If the client does not use this function, then a session is automatically started on the specified server when one of the other @aGlance/IT client functions is invoked.

Table 4-2. Supported @aGlance/IT API Functions

Function Type	API Function	OIS 40 Series/ Conductor VMS Server Support
Error handling	AAG_GetErrorText	•
Session control	AAG_StartSession	•
	AAG_EndSession	•
	AAG_Shutdown	•
Real time data - read	AAG_GetList	•
	AAG_GetTable	•
	AAG_Monitor	•
	AAG_NextMonitor	•
	AAG_MonitorArgs	•
Real time data -write	AAG_PutList	•
	AAG_PutTable	•
Historical data -read	AAG_GetHistEvent	•
	AAG_GetHistory	•
	AAG_GetHistStat	•
Historical data - write	AAG_PutHistory	Not supported
Tag data and attributes	AAG_GetTags	•
	AAG_GetAttrs	•
User task activation	AAG_ClientInit	•
	AAG_ClientTerminate	•
User commands	AAG_Command	•

Table 4-2. Supported @Glance/IT API Functions (continued)

Function Type	API Function	OIS 40 Series/ Conductor VMS Server Support
Monitor start/stop	AAG_CancelMonitor	•
	AAG_Monitor	•
	AAG_MonitorInit	•
	AAG_MonitorTerminate	•

CLIENT LISTS

The @aGlance servers can list client applications currently connected to the server. The list provides information for each client including name, address, connect time, and last activity time. A more detailed list for an individual client is available to display the number of calls to each API function. This feature is implemented using the **AAG_Command** API function to define a server specific command. The server specific command that a client can use to list the currently connected clients is **ListClients**. It optionally accepts one input value, the **ClientNumber**, which specifies the client activity to output in detail. The output is an array of strings. If no **ClientNumber** is specified, then the activity of all clients is output. For example, if using Excel:

```
=AAG ("Command", "servername", "ListClients", Sheet1!A1, Sheet1!B1:B10)
```

Where *Sheet1!A1* refers to the spreadsheet cell containing the input (**ClientNumber**) and *Sheet1!B1:B10* refers to the spreadsheet cells for the output.

The format of the **ListClients** output is illustrated by the following example. There will be a single header line and a single line for each client, up to a maximum of five clients.

Server Name	Node	Type	Version	Start Time
EBOIS	Bart	OIS	1.000	29-MAY-1996 08:07:30
Client Node	User Name	Connect Time	Last Activity Time	Function Count
NODE1		29-MAY-1996 14:50:34	29-MAY-1996 14:57:20	20

The format of the **ListClients** output using the **ClientNumber** input option is illustrated in the following example. There are eight lines of information per client.

Server Name	Node	Type	Version	Start Time
EBOIS	Bart	OIS	1.000	29-MAY-1996 08:07:30
Client Node	User Name	Connect Time	Last Activity Time	Function Count
NODE1		29-MAY-1996 14:50:34	29-MAY-1996 14:57:20	20

Server Name	Node	Type	Version	Start Time
EBOIS	Bart	OIS	1.000	29-MAY-1996 08:07:30
Client Node	User Name	Connect Time	Last Activity Time	Function Count
Activity Counts				
GetTags	2	GetAttrs4	GetList 0	
GetTable	4	PutList0	PutTable0	
Monitor	0	MonitorCancel0	MonitorUpdate9	

NOTE: In Excel, the output value cells should be set to a fixed width font such as Courier so that the columns align.

GET TAG ATTRIBUTE NAMES

The attribute list is documented as two separate lists, primary and secondary attributes. The secondary attributes are those that are not normally needed by an application. By default, only the attributes in the primary attribute list are returned by the **GetAttrs** command. This limits the number of attributes returned. To cause **GetAttrs** to return secondary attributes use the **SetOptions** command with the **AllAttrs** input string. Refer to **SET OPTIONS** for information on how to use the **SetOptions** command. If a secondary attribute is specified in a **GetList** or **GetTable** call, its value will be returned, independent of whether or not the return secondary attributes option is enabled.

SET OPTIONS

The **SetOptions** command is implemented using the server-specific command, **AAG Command**. The **SetOptions** command allows a client to modify the default behavior of the server. Currently, this function can only modify some attribute type information and monitoring behavior in the server. Table 4-3 lists the available options (arguments) for the **SetOptions** command.

NOTE: The **SetOptions** command only sets the server functionality with respect to the current client.

Table 4-3. Set Options Arguments

Option	Description
AllAttrs	Enables the server to return hidden attributes.
NoAllAttrs	Disables the server from returning hidden attributes.
PolledMonitoring=(int)	Specifies that the server should handle monitoring requests from this client where the monitor type is not explicitly set in the monitor request as being polled type. Where (int) is the polling interval in seconds. A value must be entered for (int), otherwise no polling will occur.

Table 4-3. Set Options Arguments (continued)

Option	Description
NoPolledMonitoring	Specifies that the server should handle monitoring requests from this client where the monitor type is not explicitly set in the monitor request as being exception type.

The options are specified as arguments to the **AAG_Command** function call as follows:

AAG_Command (*server, command, in_values, in_count, out_value, out_count*)

Where:

char * <i>server</i>	Name of the server.
char * <i>command</i>	Server specific command name, " SetOptions ".
AAG_ValueList <i>in_values</i>	Options listed in Table 4-3.
int <i>in_count</i>	Count of input values in the <i>in_values</i> list.
AAG_ValueList <i>out_values</i>	Not used.
int <i>out_count</i>	Not used.

The return is:

AAG_Success **SetOptions** command processed.

To change the monitoring behavior of the server for the current client from Excel, the **SetOptions** command must be invoked using a DDE Advise message similar to the following:

```
=AAGDDE|YourServer!',Command, SetOptions,
  PolledMonitoring=int,1'
```

Where:

<i>YourServer</i>	name of the server whose monitoring behavior is being modified.
<i>int</i>	Polling interval that is set in the Polled-Monitoring option. An integer must be entered for polling to occur.

NOTES:

1. A return value of **1** is required (as shown in the example DDE Advise message).
2. The last digit of the DDE Advise message command is a vertical line, not the letter **I**.

The **SetOptions** command must be invoked from the same client that is used for monitoring. Monitoring from Excel must be done using DDE Advise messages. These messages communicate via DDE with the local @aGlance/IT DDE Bridge application, which is a separate @aGlance/IT client application. When using the macro interface or the Macro Tool from Excel, Excel itself is the @aGlance/IT client application.

CALLING SECONDARY ATTRIBUTES

The attribute list is documented as two separate lists, primary and secondary. The secondary attributes are those that are not as likely to be needed by an application. By default, only the attributes in the primary attribute list are returned by the **AAG_GetAttrs** function. This limits the number of attributes returned. To cause the **AAG_GetAttrs** function to return secondary attributes, use the **AAG_Command** server-specific **SetOptions** command with the **AllAttrs** option. If a secondary attribute is specified in a **AAG_GetList** or **AAG_GetTable** call, its value will be returned, independent of whether or not the option to return secondary attributes is enabled.

GET REALTIME TAG DATA

The **Keep All Tags Connected** option should be enabled when requesting data via the **AAG_GetList** and **AAG_GetTable** functions. This option is enabled (set to yes) via the general parameters screen. Refer to the appropriate configuration instruction listed in Table 1-2 for information on the general parameters screen. If this option is not enabled then the console or workstation connects the tags and requests current values, waits up to five seconds, then returns the values.

When *No Data* is returned as a value, the corresponding item in the status list, if supplied, contains a message indicating the reason that no data was returned. The functions return **AAG_Success** even if the returned values are *No Data*. It is recommended that a status list be used in all calls that support them.

If the **Display Last Good Value** option is enabled, then the quality attribute (EBG_QUAL) should always be requested with the tag value. Refer to **ALARM QUALITY OPTIONS** in the appropriate configuration instruction listed in Table 1-2 for information on setting the **Display Last Good Value** option. Refer to [Appendix A](#) for information on the EB_QUAL attribute.

PUT REALTIME TAG DATA

Table 4-4 lists common writable attributes for all tag type and the values that can be written for the attributes. These attributes are console level write operations and only affect the console on which the @aGlance/IT server is operating. Table 4-5 lists writable attributes that are specific for each tag type. These attributes are module level write operations and affect all consoles on the communication loop. Refer to [Appendix A](#) for a complete listing of Elsag Bailey tag attributes.

Table 4-4. Writable Attributes¹, Console Level

Tag Type	Attribute	Description	Data Type	Values ⁵
ANALOG/ INTANG	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
	EB_SUBST ²	Value is substituted value	Boolean	0 = not substituted 1 = substituted
DAANLAG, DAANG, RMSC	C_APV	Analog value	Float	Expressed in engineering units
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
DADIG, DADIGTL, DD, DIGITAL, RCM, RMCB	C_DPV	Digital state	Boolean	0 = zero state 1 = one state
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
DEVSTAT	EB_SUBST ²	Value is substituted value	Boolean	0 = not substituted 1 = substituted
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
MSDD	C_IPV	Last good state	Integer	0 - 3
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
	EB_SUBST ²	Value is substituted value	Boolean	0 = not substituted 1 = substituted

Table 4-4. Writable Attributes¹, Console Level (continued)

Tag Type	Attribute	Description	Data Type	Values ⁵
STATION ⁴	C_APV	Analog value	Float	Expressed in engineering units
	C_AOUTPUT	Control output (CO)	Float	Expressed as % of span
	C_ASP	Analog set point (SP)	Float	Expressed in engineering units
	C_RATIO	Ratio index value (RI)	Float	0.0 if station is not in ratio mode
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
	EB_SUBST ²	Value is substituted value	Boolean	0 = not substituted 1 = substituted
TEXT	EB_BLINK	Text blink enabled flag	Boolean	0 = no blink 1 = blink
	EB_COLOR	Text color	Integer	0 = black 1 = white 2 = red 3 = green 4 = blue 5 = cyan 6 = magenta 7 = yellow 8 = orange 9 = yellow green 10 = green cyan 11 = cyan blue 12 = blue magenta 13 = magenta red 14 = dark gray 15 = light gray
	EB_MSG	Message number	Integer	1- 10000
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
	EB_SUBST ²	Value is substituted value	Boolean	0 = not substituted 1 = substituted
	TEXTSTR	EB_ALMSTS	Current alarm status	Integer
EB_TXTSTR		Text string data	String 80	Logic state descriptor for one state
EB_MANINHB		Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
EB_SCANSTS ³		Scan status	Boolean	0 = off scan 1 = on scan
EB_SUBST ²		Value is substituted value	Boolean	0 = not substituted 1 = substituted

Table 4-4. Writable Attributes¹, Console Level (continued)

Tag Type	Attribute	Description	Data Type	Values ⁵
INTANG	C_APV	Analog value	Float	Expressed in engineering units
	EB_ALMSTS	Current alarm status	Boolean	0 = no alarm 1 = alarm
	EB_QUAL	Quality	Boolean	0 = good 1 = bad
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan
INTDIG	C_DPVS	Digital state	Boolean	0 = zero state 1 = one state
	EB_ALMSTS	Current alarm status	Boolean	0 = no alarm 1 = alarm 4 = high alarm 5 = low alarm
	EB_QUAL	Quality	Boolean	0 = good 1 = bad
	EB_MANINHB	Alarm manually inhibited flag	Boolean	0 = not manual inhibit 1 = manual inhibit
	EB_SCANSTS ³	Scan status	Boolean	0 = off scan 1 = on scan

NOTES:

1. Some of the write operations are not instantaneous, especially the module level write operations. Therefore, it may take awhile for the console to receive the changed value in an exception report.
2. To place a tag into substitution, the EB_SUBST attribute must be written with a value of one and a substitutable attribute must immediately follow it in the same request.
3. When an on scan is issued, the client must verify the action has completed before attempting another action (i.e., control). If control is attempted immediately after an on scan, the control values are assumed to be substituted values, resulting in an already established error.
4. The STATION tag must be in computer level before the mode or any value can be changed. The STATION level can always be set. Writing EB_CMPTRSTS = 1 and EB_STNLVL = 1 in a single call will work as long as EB_CMPTRSTS appears first in the input list.
5. If an error occurs in writing the value for a tag/attribute pair, the corresponding item in the status list, if supplied, will contain a message indicating the failure. The functions return **AAG_Success** even if an error is encountered writing the specific tag-attribute pairs. It is strongly suggested that a status list be used in all calls that support them.

Table 4-5. Writable¹ Attributes, Module Level

Tag Type	Attribute	Description	Data Type	Value
DADIG	EB_ALMSUP	Enable/disable alarm suppression	Boolean	0 = unsuppress alarms 1 = suppress alarms
	EB_FORCEXR	Force an exception report	Boolean	0 = not used 1 = force an exception report
	EB_LATCHED	Value/status latched flag	Boolean	0 = reset latch 1 = not writable
	EB_NOREPORT	Module level no report flag	Boolean	0 = off report 1 = on report
	EB_SRC	Source of the reported value	Integer	0 = user inserted 1 = alternate 2 = primary
	EB_USERST	User inserted state	Boolean	Engineering unit

Table 4-5. Writable¹ Attributes, Module Level (continued)

Tag Type	Attribute	Description	Data Type	Value
DAANG	EB_ALMSUP	Enable/disable alarm suppression	Boolean	0 = unsuppress alarms 1 = suppress alarms
	EB_FORCEXR	Force an exception report	Boolean	0 = not used 1 = force an exception report
	EB_NOREPORT	Module level no report flag	Boolean	0 = off report 1 = on report
	EB_SRC	Source of the reported value	Integer	0 = user inserted 1 = alternate 2 = primary
	EB_USERVAL	User inserted state	Float	Expressed in engineering unit
DD	C_DPV	Digital state	Boolean	0 = zero state 1 = one state
	EB_DDMODE	Device driver mode	Integer	0 = auto 1 = remote 2 = manual
MSDD	C_ISP	Requested state	Integer	1 - 3
	EB_AUTOMAN	Auto manual mode	Boolean	0 = manual 1 = auto
RCM, RCMB	C_DPV	Digital state	Boolean	0 = zero state 1 = one state
RMSC	C_APV	Analog value	Float	Expressed in engineering units
STATION	C_ASP	Analog set point (SP)	Float	Expressed in engineering units
	C_AOUTPUT	Control output (CO)	Float	Expressed as % of span
	C_RATIO	Ratio index value (RI)	Float	0.0 if station is not in ratio mode
	EB_AUTOMAN	Auto manual mode	Boolean	0 = manual 1 = auto
	EB_STNVL	Station level	Boolean	0 = local 1 = computer
	EB_STNMODE	Cascade/ratio or normal flag	Boolean	0 = normal 1 = cascade/ratio
	EB_CMPTRSTS	Computer OK status flag	Boolean	0 = unused 1 = user flag
TEXTSTR	EB_TXTSTR	Text string data	String 80	Up to 80 characters text string
	EB_ALMSUP	Enable/disable alarm suppression	Boolean	0 = unsuppress alarms 1 = suppress alarms
	EB_ALMSUP	Enable/disable alarm suppression	Boolean	0 = unsuppress alarms 1 = suppress alarms
	EB_AUTOMAN	Auto manual mode	Boolean	0 = manual 1 = auto

NOTE: 1. Some of the write operations are not instantaneous, especially the module level write operations. Therefore, it may take awhile for the console to receive the changed value in an exception report.

MONITOR REALTIME TAG DATA

Monitor requests issued by the client use the **AAG_Monitor** function call. Refer to the **@aGlance/IT User's Guide** for more information on @aGlance/IT function calls. The monitor function call and parameters are:

AAG_Monitor (*server, tag_list, attr_list, list_count, value_list, [status_list], [monitor_type], [interval], [monitor_args], [arg_count], [client_data], [monitor_id]*)

Where:

char <i>*server_name</i>	Name of the server.
AAGNameList <i>tag_list</i>	Array of pointers to the tag names of the data points to be monitored.
AAGNameList <i>attr_list</i>	Array of pointers to attribute names, this list must match the number of tag-names.
int <i>list_count</i>	Number of tag/attribute pairs in the tag and attribute lists.
AAGValueList <i>value_list</i>	Array that contains the initial values for each tag/attribute pair.
AAGStatusList <i>status_list</i>	Optional parameter, array that contains the initial status for each value.
char <i>*monitor_type</i>	Optional parameter, indicates the type of monitoring to be done. Use AAG_POLLED for polled type monitoring, AAG_EXCEPTION for exception type requests.
char <i>*interval</i>	Optional parameter, monitoring time interval (for polled type requests). String represents number of seconds in the interval (i.e., 10 = 10 seconds).
AAGValueList <i>monitor_args</i>	Optional parameter, client specific argument list.
int <i>arg_count</i>	Optional parameter, number of items in the monitor arguments list.
AAGULong <i>client_data</i>	Optional parameter, client specific data that can be associated with this monitor request.
AAGULong <i>monitor_id</i>	Value generated by @aGlance/IT API which uniquely identifies the monitor request.

Real time tag data can be monitored in one of two ways, by exception or by polling. Specify the type of monitoring to be done by setting a parameter in the monitor request (**AAG_Monitor** function) from the client to the server, or by setting a monitoring type option in the server for the particular client using the **SetOptions** command.

The minimum valid polling interval supported by the server is two seconds. If polled monitoring has been specified for the client but no polling interval has been defined, the server will default the poll interval to 10 seconds.

Current limits on the server monitoring capability on OIS 40 Series consoles and Conductor VMS workstations are as follows:

- A maximum of 50 monitor requests can be active per server.
- A maximum of 100 tag/attribute pairs can be specified in a single monitor request.
- A maximum of 250 unique tags total can be monitored at one time by the server. For example, if several attributes of a single tag are being monitored, it only counts as one tag against the monitored tag total.

NOTE: Excessive loading can occur if a large number of tag/attributes are monitored in a large number of monitor requests.

If an on-line tag configuration change is made to a tag currently being monitored, the configuration changes will not always be reflected in the monitored data. If the tag configuration change affects the tag static configuration information (i.e., alarm group, logic state or engineering unit descriptors, etc.), then these changes will eventually be reflected in the monitored data for both polled and exception type monitoring. It is possible that it may take these changes longer to appear in exception monitored data than polled data, depending on overall system load.

If the on-line configuration change affects the tag name, tag descriptor, or customer tag ID of the tag, then these changes will be picked up by polled monitoring only. If it is important that all configuration changes be picked up for any tags being monitored, All clients should stop monitoring the configured tag then initiate monitoring of that tag. Note that if the on-line tag configuration change included a change in the tag type, then some attributes may begin to report a bad type status.

@aGlance/IT Server Performance

The Elsag Bailey @aGlance/IT server can support loads of providing 500 tag/attribute pair values per second to @aGlance/IT clients. This load rate should not be exceeded so that the console performance is not compromised. Additionally, the total request rate for calling @aGlance/IT functions should not exceed one request per second for each client. The limit of 500 tag/attribute pair values per second includes values that are retrieved using the monitor function.

The limit of 500 tag/attribute pair values per second is shown in the following examples. For example:

One client making one request per second for 500 tag/attribute values.

or

Five clients each requesting 100 tag/attribute pair values per second.

or

Four clients each making one request per second for 100 tag/attribute pairs values and one client monitoring 200 values which are polled every 2 seconds.

Requests for history data and event data can take considerable time for the @aGlance/IT server to retrieve. Therefore, these data types should not be requested at a high rate because they are retrievable for some time before they are no longer on-line.

When possible, monitoring should be done as exception type monitoring. Exception type monitoring performs more efficiently than polled type monitoring since no updates are sent to clients unless a tag being monitoring has changed.

When formulating monitor requests to the server, fewer monitoring requests containing more tag/attribute pairs should be used where possible. There is a notable performance improvement in both the server and the client (especially Microsoft Excel clients) when few monitor requests containing many tag/attribute pairs are used as opposed to many monitor requests containing only a few tag/attribute pairs each.

@aGlance/IT Server

When using the @aGlance/IT server to monitor data from Microsoft Excel, the values returned as *bad quality* or *no data* by the server are displayed as 0 in the Excel spreadsheet. When the server receives a monitor cancellation request from a client, it suspends sending updates to that client for any other

monitor requests that are still outstanding. This update suspension period expires in three seconds, after which updates will resume for that client provided no other monitor cancellation requests are received from it. This gives the server priority to handling cancellation requests from the client instead of sending updates to the client that could possibly saturate its buffers.

GET HISTORICAL TAG DATA

Historical tag data corresponds to the console or workstation trend data. The trending subsystem supports both standard trends (function codes 66 and 78) and enhanced trends (function code 179). Trend data can be collected in the following modes: Sample, Average, Minimum, Maximum, Sum, and Range (enhanced trends only). Standard trends have a resolution of either one minute (normal) or 15 seconds (fast). Enhanced trends have a resolution of one second or greater.

Since a single tag can be trended more than once, the tag name alone cannot be used to uniquely identify a trend. Tags such as Station tags have multiple values that can be trended, such as PV or SP. It is possible to define more than one standard trend for a single tag-mode-subtype combination. It is also possible to define both standard trends and an enhanced trend for a single tag-mode-subtype combination. However, it is not possible to define two enhanced trends for a single tag-mode-subtype combination. If two different collection periods are desired, use two tags to implement it. To uniquely identify a trend, the following information is needed:

Tag Name	Determines tag index and type.
Tag Subtype	For a station tag subtypes are PV, CO, SP, or RI.
Trend Mode	Sample, average, minimum, maximum, sum, or range.
TrendType	Normal, Fast, or Enhanced.

Table 4-6 shows the historical attribute names and the valid trend types for each suffix. There are two trend types: Standard (normal or fast), and Enhanced. Suffixes indicate the trend mode. Valid suffixes are:

- SMP, sample.
- AVE, average.
- MIN, minimum.
- MAX, maximum.
- SUM, sum.
- RNG, range.

NOTE: If the suffix is omitted, the trend mode defaults to SMP (sample).

The @aGlance/IT server imposes a maximum limit of 10 trends per request for the **GetHistory**, **GetHistEvent**, and **GetHistStat** functions. For enhanced trends, the **GetHistEvent** function will not poll a module for data. Only events stored on disk will be returned. All other functions will poll a module for data for both standard and enhanced trends.

When *No Data* is returned as a value, the corresponding item in the status list, if supplied, will contain a message indicating the reason that no data was returned. The functions return **AAG_Success** even if the returned values are *No Data*. It is strongly recommended that a status list be used in all calls that support them.

Table 4-6. Trend Types for Historical Attributes

Attribute	Trend Types by Suffix					
	SMP	AVE	MIN	MAX	SUM	RNG ¹
C_APV ²	Both	Both	Both	Both	Both	Enhanced
C_AOUTPUT ³	Both	Both	Both	Both	Both	Enhanced
C_ASP ³	Both	Both	Both	Both	Both	Enhanced
C_RATIO ³	Both	Both	Both	Both	Both	Enhanced
C_DPV ⁴	Both	Both	Both	Both3	Both	NA
C_IPV ⁵	Both	Both	Both	Both	Both	NA

NOTES:

1. The trend block must be specified as range mode (specification 3, trending mode = 32).
2. Analog tag types (i.e., ANALOG, STATION, DANG, etc.).
3. STATION tags only.
4. Digital tag types (i.e., DIGITAL, RCM, DADIG, etc.).
5. MSDD tags only.

The Elsag Bailey @aGlance/IT server may require some time to process a historic data request. The default time-out in the **AAG.INI** file may need to be increased if the client application experiences @aGlance/IT time-out errors while calling these functions. Refer to the @aGlance/IT user's guide listed in Table 1-2 for the client platform for details on increasing the time-out period.

Because of the sequential handling of requests by the @aGlance/IT software, the Elsag Bailey @aGlance/IT server can only process a single request at a time. Thus, requests that take time to complete, such as requests for historical data, can significantly delay the completion of other requests which normally complete quickly, such as requests for real-time data. They can also delay the updates of monitor data. Therefore, consider licensing multiple Elsag Bailey @aGlance/IT servers and designating one or more of them for real time use only. The servers can be uniquely named from the console or workstation, allowing clients to connect to a particular server. Refer to **@aGlance/IT Configuration** in the appropriate configuration instruction listed in Table 1-2.

AAG_GetHistory

When using the **AAG_GetHistory** function, the requested interval must be a multiple of the trend collection resolution. If it is not, no data will be returned for the trend. If data is requested at an interval less than the trend collection resolution, no data is returned. This phenomena applies to both standard and enhanced trends. For standard trends, the start time is adjusted, if necessary, to land on a boundary of the requested interval.

NOTE: The **AAG_GetHistory** function may return interpolated data.

The **AAG_GetHistory** function and parameters are:

AAG_GetHistory (*server*, *tag_list*, [*attr_list*], *hist_count*, *start_time*, *interval*, *count*, [*time_list*], *value_table*, [*status_table*])

Where:

char * server	The name of the server.
AAGNameList <i>tag_list</i>	A list of character string pointers to the tag names.
AAGNameList <i>attr_list</i>	A list of character string pointers to the attribute names. Each entry in the <i>tag_list</i> may have an attribute associated with it. This list must be the same length as the <i>tag_list</i> . C_APVAVE is an example of an <i>attr_list</i> entry. It consists of the concatenation of the INFI 90 tag attribute name (C_APV) and the trend mode (AVE) (refer to Table 4-6). If no trend mode suffix is appended to the attribute, the trend mode defaults to SMP.
int hist_count	The number of data points for which historical values are being requested. This is the number of entries in the <i>tag_list</i> .
char * start_time	The <i>start_time</i> is a string in the form <i>dd-mmm-yyyy hh:mm:ss</i> .
char * interval	The time interval in seconds represented as a string (e.g., 15 for a 15 seconds interval).
int count	The number of time intervals for which values are requested.
AAGTime <i>time_list</i>	Supplied by the server, a list of times associated with each historical record. The number of list entries is the number of requested time intervals.

AAGValueList <i>value_table</i>	The server returns the historical values for the specified data points in this table.
AAGStatusList <i>status_table</i>	An output that is an array of structures containing a status for each value. This status consists of a severity code and a pointer to error text if the severity code is not AAG_SUCCESS_SEVERITY .

The **AAG_GetHistory** function returns the historical values for a given list of tag and attribute names. The historical time range is specified as a start time, an interval time, and a number of intervals. For each time interval, a value for each of the tag/attribute pairs is returned. In a sense, a single call to **AAG_GetHistory** is similar to call to **AAG_GetList** at each historical time.

For the *m* intervals and *n* tags (and attributes), *m* by *n* values are returned. The value entries are arranged in time order. This means that the values for all the tag/attribute pairs at the first time interval appear first in the list.

AAG_GetHistEvent

The **AAG_GetHistEvent** function allows a client to retrieve compressed historical data from a server. Unlike the **AAG_GetHistory** function, the client does not specify a time interval for receiving interpolated data. Instead, the client specifies a start and end time, and the server returns historical values as they are recorded by the data historian (i.e., compressed and without interpolation).

The client specifies a list of tag and attribute names for the data points of interest. The server returns a table of values and status information. A compressed history sample may contain data for some, but not necessarily all of the requested tags. Values that are not found are returned as the value type **AAG_NODATA**.

The number of samples for a given period of time is unknown to the client, therefore an index is used to indicate whether there is more data than would fit in the allocated value and status tables. Before the first call, the client sets the index to zero to retrieve the first samples. The server updates this index so that on subsequent calls, the next batch of samples are returned.

For enhanced trends the actual raw compressed data is returned. If no events exist in the time range, then the times, data, and status values are **AAG_NODATA**. For standard trends, this function simply returns the data at the collection resolution of the trend. The time range is adjusted if necessary

for standard trends to land on a boundary for the collection resolution.

The **AAG_GetHistEvent** function and parameters are:

AAG_GetHistEvent (*server, tag_list, [attr_list], tag_count, start_time, end_time, sample_count, [time_list], value_table, [status_table], [time_index]*)

Where:

char * server	The name of the server.
AAGNameList <i>tag_list</i>	A list of requested tag names.
AAGNameList <i>attr_list</i>	A list of requested attribute names. C_APVAVE is an example of an <i>attr_list</i> entry. It consists of the concatenation of the INFI 90 tag attribute name (C_APV) and the trend mode (AVE) (refer to Table 4-6). If no trend mode suffix is appended to the attribute, the default trend mode is SMP.
int tag_count	The number of entries in the tag and attributes list.
char * start_time	The start time is a string in the form <i>dd-mmm-yyyy hh:mm:ss</i> .
char * end_time	The end time is a string in the form <i>dd-mmm-yyyy hh:mm:ss</i> .
int count	The number of time intervals for which values are requested.
int sample_count	The number of entries to be returned. This is the number of entries in <i>time_list</i> , and the number of rows or samples in the value and status table.
AAGTimeList <i>time_list</i>	Returned by the server, a list of times specifying when a sample was recorded.
AAGValueList <i>value_table</i>	A table of values returned by the server. The values fill the table based on row order (i.e., a row contains the tag values for a given time and is stored consecutively, followed by the sample for the next later time. Note that if there is no recorded value for a tag at the sample time, then the corresponding value entry will have the type AAG_NODATA .

AAGStatusList <i>status_table</i>	The server returns a status value for each entry in <i>value_table</i> . It outputs an array of structures containing a status for each value. The status consists of a severity code if the severity code is not AAG_SUCCESS_SEVERITY .
AAGULong <i>time_index</i>	Unsigned index indicating if there is more data. Initialized to zero by the client and updated by the server.

AAG_GetHistStat

The **AAG_GetHistStat** function returns statistical data, such as the minimum or average value over a time range. The statistics are calculated using the data at the collection resolution. For standard trends, raw data is used. For enhanced trends, interpolated data is used. If the enhanced trend is a sample mode trend, the data will be interpolated based on the input sampling time (FC 179, S6). If the enhanced trend is a statistical type trend, the data is interpolated based on the statistical time base (FC 179, S9 and S10). Both standard and enhanced trends have the time range adjusted if necessary to land on a boundary for the collection resolution of the trend.

The OIS 40 Series and Conductor VMS trending subsystem cannot calculate statistical data for large time spans that comprise both data saved to disk and data polled from a module. When using this function, the statistics are based only on the data polled from the module and will have a status of *Stats Suspect* if the module does not contain the data for the entire time span. If the time span encompasses only data on the disk, then that data (from the disk) is used and the module is not polled.

The **AAG_GetHistStat** function and parameters are:

AAG_GetHistStat (*server, tag_list, [attr_list], tag_count, start_time, end_time, type_count, value_table, [status_table]*)

Where:

char * server	The name of the server.
AAGNameList <i>tag_list</i>	A list of requested tag names.

AAGNameList <i>attr_list</i>	A list of requested attribute names. C_APVAVE is an example of an <i>attr_list</i> entry. It consists of the concatenation of the INFI 90 tag attribute name and the trend mode (refer to Table 4-6). The default trend type is SMP if no trend mode is appended to the attribute.
int tag_count	The number of entries in the tag and attributes list, also equal to the number of columns in the value and status tables.
char * start_time	The start time is a string in the form <i>dd-mmm-yyyy hh:mm:ss</i> .
char * end_time	The end time is a string in the form <i>dd-mmm-yyyy hh:mm:ss</i> .
int type_list	A list of character string pointers containing the statistical types. These types may be unique for certain servers. The following standard types are defined in AAGDEF.H and should be used if possible: AAG_HIST_MIN , AAG_HIST_MIN , AAG_HIST_AVE .
int type_count	The number of entries to type_count.
AAGValueList <i>value_table</i>	A table of values returned by the server. The value and status tables are ordered differently than those in the AAG_GetHistory and AAG_GetHistEvent functions. Here a table row contains all statistical types for a given tag/attribute. Tables are stored in row order. The statistical types for a given tag are stored consecutively and followed by the types for the next tag.
AAGStatusList <i>status_table</i>	If this parameter is nonzero, the server should return a status for each entry in <i>value_table</i> . It outputs an array of structures containing a status for each value. The status consists of a severity code, and a pointer to error text if the severity is not AAG_SUCCESS_SEVERITY .

GET EVENT LOG DATA

The Elsag Bailey @aGlance/IT server supports reading the system event log which includes alarms, events, and operator actions. Only events saved to the hard disk drive and within the event log are accessible. Events within the actions log are not accessible. Refer to information on system events log configuration in the appropriate configuration instruction listed in Table 1-2.

The **AAG_Command** function is used to retrieve this server-specific event data. Event data can be retrieved by time span or by a reference time and time direction. By default, the complete event message satisfying the time span or reference time and time direction is returned. Optionally, various event attributes can be selected for retrieval in place of or in addition to the complete event message.

When retrieving by time span, an optional start index parameter can be specified. This allows a client to retrieve all events for a time span than can fit in one reply. A *next index* is returned to be used as the start index for the next request if there is more data to return. When retrieving by reference time and time direction, only the event data that can fit in the reply is returned.

An event log data request may take awhile for the @aGlance/IT server to process. The default time-out in the **AAG.INI** file can be increased if the client application experiences @aGlance/IT time-out errors while using the **AAG_Command** function. Refer to the **@aGlance/IT User's Guide** for the client platform for details on increasing the time-out period.

Because the @aGlance/IT API sequentially handles requests, the Elsag Bailey @aGlance/IT server can only process a single request at a time. These requests that may take awhile to complete can significantly delay the completion of other requests which normally complete quickly, such as requests for real time data. They can also delay updates of monitor data. For this reason, consider licensing multiple Elsag Bailey @aGlance/IT servers and designating one or more of them for real time use. The servers can be uniquely named from the console or workstation, allowing clients to connect to a particular server. Refer to the appropriate configuration manual listed in Table 1-2 and **@aGlance/IT User Task Definition** for details on naming servers.

To retrieve event data, the client uses the **AAG_Command** function. The following example shows the C programming syntax for the **AAG_Command**.

AAG_Return Command (*server, command, [in_values], [in_count], [out_values], [out_count]*)

Where: .

char * server	The name of the server that will receive the command.
char * command	The server-specific command being requested, “ GetEvents ”.
AAGValueList <i>in_values</i>	A list of values to be sent to the server. Table 4-7 lists the ordered input parameters that <i>in_values</i> must contain. These parameters determine the time span retrieval case.
int in_count	The number of input values (<i>in_values</i>) being sent.
AAGValueList <i>out_values</i>	A value list to store returning data from the server.
int out_count	The number of entries allocated in the output value list (<i>out_values</i>)

Table 4-7. Time Span Retrieval Parameters

Parameter	Description
start_time	A string specifying the start time, oldest time in the standard format <i>dd-mmm-yyyy hh:mm:ss</i> . Note that this is not the same as the OIS 40 Series and Conductor VMS time format, the format is fixed and configurable text for the month component is not supported. A special value, CURRENT, designates the current time.
end_time	A string specifying the end time, most recent time in the format <i>dd-mmm-yyyy hh:mm:ss</i> . The end time must be equal to or more recent than the start time.
start_index	An optional parameter, this is an integer index indicating where in the list of matching events to begin. Initially this value should be set to 0, which indicates to start from the first matching event. If there are more events in the time period specified than can fit in the output values then the first output value will be set to a non-zero value and used as the <i>start_index</i> on subsequent calls. This enables retrieval of all events within the time period even if the requested number of output values in a single cell is too small. The first output value is set to zero when there is no more data to return beyond this call.

Several optional input strings are recognized that select which attributes of the event data to return. By default, the text of each event is returned but this can be overridden by specifying other attributes to return. Each attribute is a separate string parameter following either the *end_time* parameter or the optional *start_index* parameter when used. The order these attributes appear within this input values list is the order they

shall appear in the output values list per event returned. Up to 12 attributes can be specified at one time. Table 4-8 describes the attributes that can be selected.

Table 4-8. Returnable Attributes

Attribute Names	Description	Return Values
EB_EVENTTIME	The time of the event.	A string in the standard time format.
EB_EVENTTEXT	The text of the event.	Two strings. Since each event can be a maximum of two lines, two output values are used for each event. If a given event has only one line of text then the second output value for the event is an empty string.
EB_EVENTTYPE	The type of the event.	A string by default. Append <code>_N</code> to the attribute name to have the numeric value returned. 0 = State change and alarm 1 = Analog alarm 2 = Operator control 3 = Operator note 5 = State change 6 = Digital alarm 7 = Operator configuration 8 = Alarm acknowledge 254 = Title
EB_TAGTYPE	The type of the tag associated with the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Append <code>_N</code> to the attribute name to have the numeric value returned.
EB_TAG	The name of the tag associated with the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	String 14.
EB_ALMSTS	The alarm status of the tag associated with the event at the time of the event at the time of the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Append <code>_N</code> to the attribute name to have the numeric value returned. Refer to Table A-1 for values.
EB_QUAL	The quality of the tag associated with the event at the time of the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Append <code>_N</code> to the attribute name to have the numeric value returned. Refer to Table A-1 for values.
EB_LOOP	The hardware loop address of the tag associated with the event at the time of the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Integer. Refer to Table A-1 for values.
EB_NODE	The hardware node address of the tag associated with the event at the time of the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Integer. Refer to Table A-1 for values.

Table 4-8. Returnable Attributes (continued)

Attribute Names	Description	Return Values
EB_MODULE	The hardware module address of the tag associated with the event at the time of the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Integer. Refer to Table A-1 for values.
EB_BLOCK	The hardware block address of the tag associated with the event at the time of the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Integer. Refer to Table A-1 for values.
EB_ALMGRP	The alarm group of the tag associated with the event at the time of the event. If an event is not associated with a tag then AAG_NODATA value will be returned.	Integer. Refer to Table A-1 for values.

The output values of this command depend on the input values. If a *start_index* is specified then the first output value will be a numeric *next_index* that can be used as the *start_index* of a subsequent call to retrieve any additional events in the time period. The *next_index* value goes to zero if all events in the time period have been retrieved. The remainder of the output values are based on the event attributes requested by the input values. If no event attributes are requested then the output will be the text of the events. If specific event attributes are requested via the input values then each of those attributes will be returned for each event. As many events that can fit in the output values will be returned. Note that near the end of the output values list all the requested attributes for an event must fit for any of the attributes for the event to be returned.

The output values list (*out_values*) contains storage for the reply and must be sized to hold data for at least one event. In the default output case, where no event attributes are specified, at least two output slots (within the *out_values*) are required for a successful reply. The default output case returns complete event messages. Each event message consists of two lines of text with up to 255 characters per line. An output slot or value is required for each line of text. If the second text line is empty, its corresponding output slot is returned with **AAG_NODATA**.

Data is returned in row order. In other words, all data corresponding to an event is returned in consecutive output slots followed by data for the next event. Thus, data for the first event appears first in the output value followed by data for the second event. When event attributes are selected, they are returned in the order they appear within the input value list.

When the *start_index* option is used, *out_values* must be sized to hold at least two rows of event data. The first row is reserved for the *next_index* indicator. The second row contains the first

event. The first output slot contains the *next_index* value where a non-zero value means there is more data to retrieve. The remaining output slots within the first row contain the **AAG_NODATA** data type. Note that when the *start_index* is not used, the first row of output represents the first event.

The reply shall not contain partial event data. All unused output slots shall be returned with a **AAG_NODATA** data type. For example, when an event data requested requires three output slots, and only one output slot remains, that event data is not returned. It can be retrieved by a subsequent request if the *start_index* option is used.

All output slots contain valid data only when the **GetEvents** command returns successfully. The output slots are undefined if this command returns unsuccessfully.

Refer to Table 4-9 for input parameters that the *in_values* must contain for reference time retrieval.

Table 4-9. Reference Time Retrieval Parameters

Parameter	Description
Start time	A string specifying the start/reference time. The format is <i>dd-mmm-yyyy hh:mm:ss</i> . Reference time represents the oldest time (to return) when direction specified is FORWARD. It represents the most recent time when direction is BACKWARD.
Direction	A string specifying the direction to move with respect to the start/reference time. FORWARD means to move forward in time beginning at the start/reference time. FORWARD cannot be used in conjunction with a start time of CURRENT. BACKWARD means to move backward in time beginning at the start/reference time.

Optional attribute parameters can be specified as in the time span retrieval case. These parameters must follow the *direction* parameter. The order they appear in this input list is the order they shall appear in the output list per event returned.

Unlike the time span retrieval case (with the *start_index* option), there is no related subsequent call. This retrieval method is for retrieving a fixed number of event data near the reference time.

The data returned is in row order. The first row represents the event closest to the start time followed by the next event in the time direction specified. All relevant output restrictions mentioned in the time span retrieval case are followed. Figures 4-1 and 4-2 show an example of how the time span retrieval and reference time retrieval parameters are used in the **GetEvents**

command. Refer to [Section 5](#) for tables listing applicable error codes for this function.

GetEvents		
Input Values	Output Values	
10-MAY-1996 10:20:30	28947	Next start index
10-MAY-1996 12:20:30		Pad to start of events
0	10:20:31 A100 109.2H TANK 1 LEVEL	Line 1 of first matching event
	TOO FULL	Line 2 of first matching event
	10:20:45 D100 OPEN VALVE 4	Line 1 of second matching event
		Line 2 of second matching event, empty
	10:21:05 A200 106.9 GAS FLOW	Line 1 of third matching event
	OK	Line 2 of third matching event
		Empty, not enough output values remain to fit all attributes for another event

Figure 4-1. Get Events Example (Time Span Retrieval)

GetEvents		
Input Values	Output Values	
10-MAY-1996 10:20:30	10-MAY-1996 10:20:31	Time of first matching event
FORWARD	10:20:31 A100 109.2H TANK 1 LEVEL	Line 1 of first matching event
EB_EVENTIME	TOO FULL	Line 2 of first matching event
EB_EVENTEXT	10-MAY-1996 10:20:45	Time of second matching event
	10:20:45 D100 OPEN VALVE 4	Line 1 of second matching event
		Line 2 of second matching event, empty
	10-MAY-1996 10:21:05	Time of third matching event
	10:21:05 A200 106.9 GAS FLOW	Line 1 of third matching event
	OK	Line 2 of third matching event
		Empty, not enough output values remain to fit all attributes for another event

Figure 4-2. GetEvents Example (Reference Time Retrieval)

GET WINDOW INFORMATION

The Elsag Bailey @aGlance/IT server supports reading information about a specific console or workstation window. The information provided is the same that is returned by the **uti_get_crt_info()** function. Key lock state information (tune locked and config locked) and X window location (server name, server number, screen number, and network transport) are available. This feature uses the **AAG_Command** function to define a server-specific command.

The server-specific command that a client uses to retrieve information for a specific console or workstation window is **GetWindow**. It accepts one input value. Refer to Table 4-10 for the **GetWindow** input value.

Table 4-10. Input Value (GetWindow)

Input Value	Type	Description
Window	Integer, float, or string	One relative value of window number. Currently the range of the this value is 1 to 8.

There are six output values. Table 4-11 lists the order that the output values are returned. Refer to Section 5 for tables listing error codes associated with the **GetWindow** function.

The following Excel spreadsheet macro shows how this function can be used. Refer to the *@aGlance/IT User's Guide* for more information on using the **AAG_Command** function in spreadsheet macros.

```
Example =AAG ("Command", "server_name", "GetWindow", Sheet1!A1, Sheet1!B1:B6
```

Sheet1A1 is the *inputvalues* option of the command and is an Excel cell reference where a relative window value has been placed by the user. This input value is described in Table 4-10.

Sheet1!B1:B6 is the *outputvalues* option of the command and is an Excel cell reference where the user desires the six output values to be placed.

Table 4-11. Output Values (GetWindow)

Output Value	Type	Description
Server Number	Integer	X window server number on which the console or workstation window is displayed.
Screen Number	Integer	X window server screen number on which the console or workstation window is displayed.
Server Name	String	X window server name on which the console or workstation window is displayed.

Table 4-11. Output Values (GetWindow) (continued)

Output Value	Type	Description
Transport	String	Network transport (DECNET, TCPIP, LOCAL) via which the OIS 40 Series or Conductor VMS software is communicating with the X window server to display the console or workstation window.
Tune Lock State	Boolean	State of the configuration key lock associated with the console or workstation window.
Configuration Lock State	Boolean	State of the configuration key lock associated with the console or workstation window.

REQUEST DISPLAY

The Elsag Bailey @aGlance/IT server supports a request to call up a SODG generated graphic display on an active console or workstation window. The window number and display name are specified in the request. Optionally, a time-out period and acknowledgment prompt can also be specified. The prompt appears in an optional Motif dialog box that alerts the user of the request and provide the option to decline the display. The operator acknowledgment option is configurable on the console via the user task options menu. If operator acknowledgment is required, the client task and the Elsag Bailey @aGlance/IT server is suspended until the operator responds to the prompt or the time-out period expires. This feature uses the **AAG_Command** function to define a server-specific command.

A display request may take some time for the Elsag Bailey @aGlance/IT server to process if acknowledgment dialog is enabled. It may be necessary to increase the default time-out in the **AAG.INI** file if the client application experiences @aGlance/IT time-out errors when calling these functions.

Because of sequential handling of requests by the @aGlance/IT software, the Elsag Bailey @aGlance/IT server can only process a single request at a time. Requests for an acknowledgment display may take some time to complete and may significantly delay the completion of other requests which normally complete quickly, such as requests for real-time data. Requests for an acknowledgment display may also delay updates of monitor data. Therefore, consider licensing multiple Elsag Bailey @aGlance/IT servers and designating one or more of them for real time use only. The servers can be uniquely named from the console or workstation, allowing clients to connect to a particular server. Refer to **@aGlance/IT Configuration** in the appropriate configuration instruction listed in Table 1-2.

A client uses the **RequestDisplay** server-specific command to request a display. Table 4-12 list the **RequestDisplay** command input parameters. There are no output values. If an output valid list is provided it is ignored.

If the *user task display activation acknowledgment required flag* is enabled, the Elsag Bailey @aGlance/IT server will not process any requests from any clients while the acknowledgment dialog is active. Refer to [Section 5](#) for tables listing error codes associated with the **RequestDisplay** function.

Table 4-12. *Input Values (RequestDisplay)*

Input Value	Type	Description
Window	Integer, float, or string	Window number (can be a number coded as a string, but must be a number).
Display Name	String	Console or workstation display name, up to eight characters.
Timeout	Integer, float, or string	Optional input value. Time-out period for dialog box in seconds. If time-out is less than or equal to zero then no time-out is used and the command waits indefinitely on an operator response. No time-out is the default.
Prompt	Integer, float, or string	Optional input value. A prompt for the dialog box with a maximum of 60 characters. A default prompt is used if none is specified.

For information on implementing this function using Excel or Lotus 1-2-3 refer to the *@aGlance/IT User's Guide*.

WRITE TO DIAGNOSTIC LOG

The Elsag Bailey @aGlance/IT server supports requests to write console or workstation diagnostic log messages. The message is time stamped and written to the circular diagnostic log file (VLOG) on the console or workstation. The message includes the client's host name and user name. This feature is implemented using the **AAG_Command** function to define a server-specific command.

A client can use the **DiagnosticLog** command to write a message to the OIS 40 Series and Conductor VMS diagnostic log. There are no restrictions on the input value list except that it must contain at least one value. Table 4-13 contains an example of a **DiagnosticLog** command input value. Numeric input values are converted to strings before being written to the console or workstation diagnostic log. There are no output values. If an output valid list is provided it is ignored. Refer to [Section 5](#) for tables listing error codes associated with the **DiagnosticLog** function.

Table 4-13. *Input Value (DiagnosticLog)*

Input Value	Type	Description
Value1..n	Integer, float, or string	Information to be written to the OIS 40 Series and Conductor VMS diagnostic log.

ACTIVATE REMOTE TASK

The Elsag Bailey @aGlance/IT server supports sending an activation command to a remote @aGlance/IT server. The activation request contains activation parameters. The user task activation is performed using a touch point (i.e. mouse, trackball, or touch screen), key select, function key, or annunciator display panel (ADP) push-button.

This functionality is intended to mimic the user task interface (UTI) user task activation functionality. With the user task interface, the user task activation mechanism can run a user task on a remote VMS hosted computer that is running DECnet and has the UTI environment installed. With an @aGlance/IT server, the user tasks can run on a variety of platforms and communicating over a variety of network protocols. Therefore, the @aGlance/IT user task activation command does not actually run a user task, but sends an activation command to a remote server that is already running.

The @aGlance/IT remote server activation request is implemented using the **AAG_Command** function to request a server-specific command. The implementer of the remote server is responsible for implementing the server-specific request with the proper input values listed in Table 4-14. Note that if the user-written server needs the node name of the OIS 40 Series console or Conductor VMS workstation, then it can be obtained by calling the **AAG_GetCurrentSession** function.

Table 4-14. Input Values (User Task Activation)

Input Value	Type	Description
0	Integer	Number of the console or workstation window from which the activation was initiated (1-based).
1	String	Name of the first server running on the console or workstation from which the activation was initiated. This is a string of length zero if there are no servers running on the console or workstation.
2	String	Parameter string specified by the user in the configuration of the touch point, key select, function key, annunciator display panel (ADP) push-button, or command file.

The OIS 40 Series console and Conductor VMS workstation can only process one activation request at a time because of restrictions imposed by the @aGlance/IT software. When the console or workstation sends an activation message to a remote server, no other activation requests can be processed until the remote server responds to the activation request. Certain guidelines must be followed when defining the specified command for the remote server so that the console or workstation can process multiple activation requests quickly.

The user implemented server must process the command quickly. If the command starts a long series of actions, the server must be coded so that it can provide the return to the console or workstation activation client in a timely manner. Specifically, have the server set a flag to alert the applications dispatch code and then return from the command callback. The user implemented servers dispatch code can then start processing the request. The user implemented server will need to define an **Event_Notify** callback so that the **AAG_Dispatch** function is called only when an @aGlance/IT request is received.

Touch Point Activation

@aGlance/IT user task activation via touch point is configured by the new display interactive escape:

ei 107, 116 (touch point select @aGlance/IT user task activation command)

The touch point select @aGlance/IT user task activation command defines a touch point area on the screen used to send a command to a user written @aGlance/IT server program. This command can only be used with an OIS 40 Series console or Conductor VMS workstation. The command and parameters are:

ei 107, 116, x1-coord, y1-coord, x2-coord, y2-coord, server-name~, commandname~, connection, argument~

Refer to Table 4-15 for a description of command parameters. An example of the touch point command and parameters is shown.

ei 107, 116, 1200, 5200, 1600, 5600, USER\$SERVER~, OIS_ACTIVATION~, 1, STARTCLIENT~

The console or workstation can only process one user task activation request at a time because of restrictions imposed by the @aGlance/IT software. When the console or workstation sends an user task activation message to a user's server, no other user task activation requests can be processed until the user's server has responded to the request. The server must define the specified command according to certain guidelines so that the console or workstation can process multiple user task activation requests quickly.

The user implemented server must process the command quickly. If the command starts a long series of actions, the server must be coded so that it can provide the return to the console or workstation activation client in a timely manner. Specifically, have the server set a flag to alert the applications dispatch code and then return from the command callback.

Table 4-15. Command Parameters (Touch Point)

Parameter	Description
x1-coord, y1-coord	Lower left corner coordinates.
x2-coord, y2-coord	Upper right corner coordinates.
servername~	Name of the user written @aGlance/IT server. The server name can be either a logical or actual server name. If using a logical name, it must appear in the system logical name table. Maximum of 32 characters followed by a tilde (~).
commandname~	Name of the server-specific command to invoke. Maximum of 32 characters followed by a tilde (~).
connection	0 = disconnect 1 = keep connected
argument~	A string to be passed with the command as one of the input values. Maximum of 80 characters followed by a tilde (~).

The user implemented servers dispatch code can then start processing the request. The user implemented server will need to define an **Event_Notify** callback so that the **AAG_Dispatch** function is called only when an @aGlance/IT request is received

User task activation requests for servers that are infrequently used and to not require rapid activation should use the disconnect option to save on resources. This option should also be used if the server allows only a limited number of sessions. Requests for servers that are frequently used should specify the keep connected option, which allows for faster communication on subsequent activations.

Key Select Activation

The @aGlance/IT user task activation via key select is configured by the new display interactive escape:

ei 108, 116 (key select @aGlance/IT user task activation command)

The key select @aGlance/IT user task activation command defines a key or key sequence used to send a command to a user written @aGlance/IT server program. This command can only be used with an OIS 40 Series console or Conductor VMS workstation. The command and parameters are:

ei 108, 116, *key1*, *key2*, *servername~*, *commandname~*, *connection*, *argument~*

Refer to Table 4-16 for a description of command parameters. An example of the key select command and parameters is shown.

ei 107, 116, 49, 65, USER\$SERVER~, REMOTE_ACTIVATION~, 1, STARTCLIENT~

Table 4-16. Command Parameters (Key Select)

Parameter	Description
key1	ASCII key code for the first key.
key2	ASCII key code for the second key.
servername~	Name of the user written @aGlance/IT server. The server name can be either a logical or actual server name. If using a logical name, it must appear in the system logical name table. Maximum of 32 characters followed by a tilde (~).
commandname~	Name of the server-specific command to invoke. Maximum of 32 characters followed by a tilde (~).
connection	0 = disconnect 1 = keep connected
argument~	A string to be passed with the command as one of the input values. Maximum of 80 characters followed by a tilde (~).

The console or workstation can only process one user task activation request at a time because of restrictions imposed by the @aGlance/IT software. When the console or workstation sends an user task activation message to a user's server, no other user task activation requests can be processed until the user's server has responded to the request. The server must define the specified command according to certain guidelines so that the console or workstation can process multiple user task activation requests quickly.

The user-implemented server must process the command quickly. If the command starts a long series of actions, the server must be coded so that it can provide the return to the console or workstation activation client in a timely manner. Specifically, have the server set a flag to alert the applications dispatch code and then return from the command callback. The user implemented servers dispatch code can then start processing the request. The user implemented server will need to define an **Event_Notify** callback so that the **AAG_Dispatch** function is called only when an @aGlance/IT request is received

User task activation requests for servers that are infrequently used and to not require rapid activation should use the disconnect option to save on resources. This option should also be used if the server allows only a limited number of sessions. Requests for servers that are frequently used should specify the keep connected option, which allows for faster communication on subsequent activations.

Function Key Activation

The @aGlance/IT user task activation via function key and annunciator display panel (ADP) push-button are configured as usual, through the user task definition, function keys, and ADP configuration screens on the console. The only difference is that the labels and meanings of the fields in the user task definition screen correspond to the @aGlance/IT user task activation instead of the UTI user task activation. Table 4-17 list the differences in the user task definition screen fields (i.e., the difference between a normal field and @aGlance/IT field).

Table 4-17. Screen Fields (User Task Definition)

Normal Field	@aGlance/IT Field	New Meaning
Task node name	Server name	Name of the user written @aGlance/IT server. The server name can be either a logical name or actual server name. If using a logical name, it must appear in the system logical name table.
Task pathname	Server command	Name of the server-specific command to invoke.
Task activate code	Remain connected?	This field accepts a value of YES or NO.
Task argument string	Command argument string	A string to be passed with the command as one of the input values.

Command Processor Activation

The **AAG_Command** processor task (**AAG_CP**) is a program that can be used to send commands to one or more user written @aGlance/IT server tasks. Its main purpose is to send commands during the start-up or shutdown of the console software.

During console start-up and shutdown the **AAG_CP** program is invoked with the file **user_commands.csf** in the [OIS.CONFIG] directory. This file does not exist by default and must be created. Refer to the **GEN_TSF** command in Table 4-18 for a description of how to create this file.

The following information describes the **AAG_Command** processor usage in detail. To invoke a command processor function use the syntax:

AAG_CP *command file_spec*

Where:

command A command from Table 4-18.

file_spec A command specification file that requires the complete path to the file. The file should be given the name **user_commands.csf** and must reside in the [OIS.CONFIG] directory to be used during console start-up and shutdown.

The format of a command specification record in the command specification file (**user_commands.csf**) contains the following data items separated by commas. Refer to Table 4-19 for a description of the data items in the command specification file.

start, stop, target, server, command, [parameter string]

Table 4-18. AAG_CP Commands

Command	Description
CK_CSF	Used to syntax check a command specification file.
GEN_TSF	Generates a sample command specification file. This file contains comments (any line beginning with an *) and command specification records.
HELP	Displays a brief description of the command processor commands. This command accepts no arguments (i.e., <i>file_spec</i>).
START	Used to send a command to a server when the console software is started. This command selects the command specification records in the command specification file whose <i>start</i> field contains Y. ¹
STOP	Used to send a command to a server when the console software is shutdown. This command selects the command specification records in the command specification file whose <i>stop</i> field contains Y. ¹

NOTE: 1. The CRT parameter of the command is set to zero to indicate that this is an activation by **AAG_CP** not from a console display.

Table 4-19. AAG_CP Command Specification File

Command	Contents	Description
start	Y or N	If Y, this command is sent when the console software is started. This field is case sensitive.
stop	Y or N	If N, this command is sent when the console software is shutdown. This field is case sensitive.
server	Server name	Name of the user-written server task to which to send the command. There is a 32 character limit to this field. This field is not case sensitive.
command	Command name	Name of the command to send to the user-written server task. The command must conform to the command layout guidelines contained in ACTIVATE REMOTE TASK . There is a 32 character limit to this field. It may or may not be case sensitive depending on how the command is defined in the server.
parameter string	Up to an 80 character string	Optional string to pass along with the command. This field can be left blank. It can contain commas.

SECTION 5 - TROUBLESHOOTING

INTRODUCTION

This section explains how to use the diagnostic capabilities of the @aGlance/IT server provided by the **Diagnose** server-specific command. It also contains a table of status messages and a table of return error messages. Each table lists the return text, the applicable client functions, and a description (meaning) of the return text.

SERVER DIAGNOSTICS

@aGlance/IT servers provide diagnostic capabilities to debug the current server state. This feature is implemented using the **AAG_Command** function to define a server-specific command.

Each @aGlance/IT server maintains a circular log file containing error and informational messages. By default, only errors are logged and a command is required to view the error log and to enable/disable informational message logging.

The server-specific command that a client can use to view the server log and enable/disable logging of informational message is **Diagnose**. The log is viewed from the most recent to the oldest message. To view the log file, use the **AAGLog n** function from a console or workstation, where n a log file number. Refer to the file utilities instruction listed in Table 1-2 for more information on using the **AAGLog n** function.

Several optional input values are recognized that allow for paging through the log file and enabling/disabling logging informational files. Table 5-1 lists and describes the optional input values. These strings are not case sensitive.

If an index is specified as an input value, then the first output value will be the index of the next unread message or 0 if there are no more messages in the log. The remainder of the output will be an array of message strings, each one 132 characters or less. Each string will contain the time the message was logged, the applicable node name, and user name (if available). Note that if a single message needs to be longer than 132 characters then the time and node/user name need appear only on the first line. If the input values consist only of the strings to enable/disable informational messages then the client is not required to request output values.

Table 5-1. Diagnose Command Options

Input Value	Description
Numeric index	Index of the first message to view in the log. A zero implies the most recent message.
AllClients	Enable/disable informational message logging for all clients based on the input values that follow.
[No]Informational	Enable/disable all informational messages.
[No]GetTags	Enable/disable logging for GetTags related messages.
[No]GetAttrs	Enable/disable logging for GetAttrs related messages.
[No]GetList	Enable/disable logging for GetList related messages.
[No]GetTable	Enable/disable logging for GetTable related messages.
[No]PutList	Enable/disable logging for PutList related messages.
[No]PutTable	Enable/disable logging for PutTable related messages.
[No]GetHistory	Enable/disable logging for GetHistory related messages.
[No]GetHistEvent	Enable/disable logging for GeHistEvent related messages.
[No]GetHistStat	Enable/disable logging for GetHistStat related messages.
[No]Monitor	Enable/disable logging for monitoring related messages.

The following examples show how the server-specific **Diagnose** command is used by an Excel spreadsheet add-in. Refer to the **@aGlance/IT User's Guide** for more information on the use of the **AAG_Command** function and how to use the function with a Lotus 1-2-3 spreadsheet add-in.

The **Diagnose** command options (input values shown in Table 5-1) are placed in spreadsheet cells. Then write a macro using the spreadsheet cell input values as input or output options for the server-specific **Diagnose** command. The resulting macros are used to enable or disable logging of messages and to look at a log file. By default, all informational message logging is disabled.

Example 1 Macro to enable and disable diagnostics using the **AAG_Command** input values option.

```
=AAG ("Command", "yourserver", "Diagnose", Sheet1!A1:A3,)
```

This macro inputs cell values from Sheet1 where the spreadsheet cell input values enable message logging for **GetTags** and **GetAttrs** related messages, and disables message logging for **GetList**.

The Sheet1 spreadsheet containing input option values for this example:

Sheet1			
	A	B	C
1	GetTags		
2	GetAttrs		
3	NoGetList		
4			

Example 2 Macro to view latest diagnostic message log using the **AAG_Command** output values option.

```
=AAG ("Command", "yourserver", "Diagnose", , Sheet2!A1:A10)
```

This macro outputs diagnostic messages to Sheet2 cells for all diagnostic messages that have logging enabled. The output values option of the **AAG_Command** places the diagnostic messages in the output areas set up on spreadsheet Sheet2.

The Sheet2 spreadsheet containing output areas for this example:

Sheet2	
	A
1	Diagnostic message output area
2	Diagnostic message output area
3	Diagnostic message output area
4	Diagnostic message output area
5	Diagnostic message output area
6	Diagnostic message output area
7	etc.
8	
9	
10	

NOTE: In Excel, the format of the output cells should be TEXT, not GENERAL. Since the beginning of each line of output contains the date and time, the value is interpreted as a date if the cell has a format of GENERAL.

Example 3 Macro to scroll through diagnostic messages using **AAG_Command** input and output values option.

```
=AAG ("Command", "yourserver", "Diagnose", Sheet3!A1, Sheet3!B1:B10)
```

This macro inputs a numeric index from a input value cell to determine a starting point for scrolling through diagnostic messages. The output area for the diagnostic messages is setup on the Sheet3 spreadsheet then those areas are filled by the **AAG_Command** output values option.

The Sheet3 spreadsheet containing input and output option cells for this example:

Sheet3		
	A	B
1	0	Diagnostic message output area ¹
2		Diagnostic message output area
3		Diagnostic message output area
4		Diagnostic message output area
5		etc.
6		
7		
8		
9		
10		

NOTE: 1. The first output cell contains the next numeric index.

STATUS LIST MESSAGES

Table 5-2 lists @aGlance/IT server status messages that are returned in the status list of function calls that have the *status range* parameter available. It is recommended that a status list be used in all calls that support them because functions will return **AAG_Success** even if the returned values are *No Data*. The status information in the status list applies to individual tag/attribute pairs of the function, not the function itself.

Table 5-2. Status Messages

Status Text	Applicable Client Functions	Description
Bad Attribute	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor, AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	The attribute name specified is not a valid attribute name in the given context. For example, AAG_PutList will report this error for a misspelled attribute name or for an attribute that is not writable, even if it is the name of a valid readable attribute.
Bad Level	AAG_PutList, AAG_PutTable	An attempt to perform process control on a STATION tag was attempted when the station was not at COMPUTER level. The only valid process control operations for a STATION that is not at COMPUTER level is to write EB_CMPTRSTS and EB_LEVEL.
Bad Quality	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor	The tag is in bad quality so the value is not available and process control is inhibited.
Bad Read	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor	Unable to read tag configuration information from file.

Table 5-2. Status Messages (continued)

Status Text	Applicable Client Functions	Description
Bad Tag	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor, AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	The tag name specified is not defined on the console on which the server is running. Note that tag names are case sensitive.
Bad Type	AAG_GetList, AAG_GetTable, AAG_Monitor	The attribute specified does not apply to the type of the given tag.
Bad Value	AAG_PutList, AAG_PutTable	The given value cannot be converted to the expected data type of the attribute. For example, a string containing non-numeric characters is specified for an integer attribute.
Bad Write	AAG_PutList, AAG_PutTable	Unable to write tag configuration information to file.
Disestablished	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor	The tag is off-scan.
Mode Interlocked	AAG_PutList, AAG_PutTable	The STATION tag is mode locked and the mode and level cannot be changed.
No Specs	AAG_GetList, AAG_GetTable, AAG_Monitor	No specifications have been received for the tag and the specified attribute is an INFI90 specification value.
Not in Substitute	AAG_PutList, AAG_PutTable	An attempt was made to write an attribute that is only substitutable, but the tag is not in substitute. The EB_SUBST attribute must be written with a value of one, directly followed by a substitutable attribute in the same request, in order to place the tag in substitute mode.
Not Processed	AAG_PutList, AAG_PutTable	The tag attribute pair was not processed due to a previous error writing an attribute for the same tag. The writing of subsequent attributes for the same tag is aborted when an error occurs to protect against the tag being in an unexpected state when writing values.
RED TAGGED	AAG_PutList, AAG_PutTable	The tag is red tagged and the <i>Suppress Control if Red Tagged</i> system configuration option is enabled so process control is inhibited.
Source Locked	AAG_PutList, AAG_PutTable	An attempt was made to change the input source for a DANG or DADIG tag, but the corresponding block is locked into the current input source.
Substituted	AAG_PutList, AAG_PutTable	An attempt was made to write a process control attribute when the tag is in substitute mode. The tag must be placed on scan before a process control operation can be performed.
TRACKING	AAG_PutList, AAG_PutTable	An attempt was made to write to a value that is being tracked by INFI90, such as setting the set point on a STATION that is set point tracking.
Trend not found	AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	No trend was found matching the given tag name and attribute.

Table 5-2. Status Messages (continued)

Status Text	Applicable Client Functions	Description
Value out of range	AAG_PutList, AAG_PutTable	The given value to write is not within the range of the expected values for the specified attribute.
Wrong Mode	AAG_PutList, AAG_PutTable	An attempt was made to perform a control operation on a STATION tag when the STATION is not in the correct mode. For example, attempting to write the control output when the STATION is in AUTO mode.
Wrong Station Type	AAG_PutList, AAG_PutTable	An attempt was made to put a STATION into RATIO or CASCADE mode but the station is not a RATIO or CASCADE station.
Resolution Invalid	AAG_GetHistory	The requested interval time (e.g., 15 seconds) is less than the collection resolution (e.g., one minute) of a trend in the request.
Not In Yet	AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	A request for data was made but the data has not been processed or received yet.
Not Connected	AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	Cannot connect to the tag associated with the trend. This occurs if the tag type has been changed, or the tag has been deleted from the tag database. It also occurs if the tag index, trend mode, or tag subtype has been changed in the trend database, or a trend has been deleted from the trend database while processing other trends in the same request.
Stats Suspect	AAG_GetHistStat	At least one value used in a statistical calculation had bad quality.
Missing	AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	Data is not available on the console. For standard trends, this normally occurs for time periods when the console or workstation is shutdown. For enhanced trends, this will depend on how long the module can store the data.
Not Retrieved	AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	Data is not currently available on the console, but is should be available due to a past archive to media. The data must be retrieved using archive data retrieval procedures.
Not Available	AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	Data is not currently available on the console and an attempt to retrieve the data from archive media was made but unsuccessful due to bad data on the archive media.
Data Error	AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	Error encountered during an attempt to read trend data from an archive media.
Internal Error	AAG_GetHistory AAG_GetHistEvent AAG_GetHistStat	Some internal processing error occurred.

RETURN ERROR MESSAGES

Table 5-3 lists @aGlance/IT server status messages that are returned in the status list of function calls who have that parameter available. The status information applies to a particular tag/attribute pair if it is enabled resulting from unsuccessful client function requests. The client can get the corresponding message by calling **AAG_GetErrorText** whenever its request function returns an **AAG_SERVER_FAILS** status.

Table 5-3. Error Messages

Return Error Text	Applicable Client Functions	Description
All attributes invalid	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor, AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	None of the attributes specified in the request are valid attribute names in the given context. For example, AAG_PutList reports this error if all attribute names are misspelled or if none of the attributes are writable, even if any of them are the name of a valid readable attribute.
All tags invalid	AAG_GetAttrs, AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor, AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	None of the tags specified in the request are defined on the console or workstation where the server is running.
Bad display name specified	RequestDisplay	The display name input value is not a string.
Bad statistical type	AAG_GetHistStat	The statistical type specified is not recognized. The valid statistical types are: AAG_HIST_MAX AAG_HIST_MIN AAG_HIST_AVG
CIU Error	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor	An error occurred when trying to read from or write to the computer interface unit, such as a time-out on a busy loop.
Client # (n) not found	ListClients	The client number specified exceeds the number of currently connected clients. The List Clients commands only lists information on currently connected clients.
Display name (<i>display</i>) invalid	RequestDisplay	The specified display was not found on the console or workstation where the server is running.
Display of <i>display</i> on window <i>n</i> ignored	RequestDisplay	The operator rejected the display callup by choosing the <i>Ignore</i> button in the acknowledgment dialog.
Display of <i>display</i> on window <i>n</i> timed out	RequestDisplay	The operator did not acknowledge the display request within the time-out period specified in the request.

Table 5-3. Error Messages (continued)

Return Error Text	Applicable Client Functions	Description
Error displaying dialog box on window <i>n</i>	RequestDisplay	An error occurred on the console or workstation when attempting to bring up the acknowledgment dialog for the display.
Error encountered; some data being returned	GetEvents	An error occurred while reading the event log snapshot file but data processed before the error will be returned.
Error sending display	RequestDisplay	An error occurred on the console or workstation when attempting to callup the display.
Event data not being saved to disk	GetEvents	The <i>System Event Log</i> configuration is not setup to save events to disk. Events must be saved to disk for the GetEvents command to function.
Forward option invalid with current or future time	GetEvents	The FORWARD option was specified with a start time for which no later events are possible.
Incorrect time format	GetEvents	The format of the start or end time is incorrect. The time must be a string in the form "DD-MMM-YYYY HH:MM:SS" or "CURRENT".
Invalid argument count	Diagnose, DiagnosticLog, GetWindow, ListClients, RequestDisplay, SetOptions	The number of input values does not conform to the requirements of the command.
Invalid client # (<i>n</i>) specified	ListClients	The client number specified is not within the range of support clients (1 - 5)
Invalid monitor type requested	AAG_Monitor	The monitor type specified is not recognized. The valid monitor types are as follows: AAG_POLLED AAG_EXCEPTION
Invalid time range	AAG_GetHistEvent, AAG_GetHistStat	The specified start time is later than the specified end time.
Invalid window # (<i>n</i>) specified	GetWindow, RequestDisplay	The window number specified is not within the range supported by the console or workstation (1-8).
Mailbox error	All	A communication error occurred on the console or workstation while attempting to perform the request.
Max tags per monitor request limit exceeded	AAG_Monitor	The request contains more than the maximum limit of 100 tag attribute pairs.
Maximum monitor request limit reached	AAG_Monitor	Each server supports up to 50 active monitor requests.
Memory allocation or system communication failure	AAG_GetList, AAG_GetTable, AAG_Monitor	There is insufficient memory on the console or workstation to perform the request or a communication error occurred on the console or workstation while attempting to perform the request.
Monitor request CB connect failed	AAG_Monitor	The maximum number of monitored tags was exceeded. Each server supports up to 250 monitored tags.

Table 5-3. Error Messages (continued)

Return Error Text	Applicable Client Functions	Description
Monitor update failed aborting monitor request	AAG_Monitor	An error occurred while attempting to update a monitor request. View the server diagnostic log for additional information.
No data match request criteria	GetEvents	No events matching the input time range or option were found.
No matching attributes found	AAG_GetAttrs	No attributes matching the given pattern from the given start index were found.
No matching tags found	AAG_GetTags	No tags matching the given pattern from the given start index were found.
Output buffer too small for request	GetEvents	The number of output values is insufficient to return the selected (or default) attributes for even a single event.
Requested monitor poll interval is invalid	AAG_Monitor	The poll interval specified in the request is less than the minimum poll interval of two seconds.
Server monitoring disabled, initialization failed	AAG_Monitor	The server encounter a memory allocation error while trying to initialize monitoring.
Shared memory allocation error	All	There is insufficient memory on the console or workstation to perform the request.
Start index no longer valid	GetEvents	The start index is no longer valid since a new event log snapshot was performed since the last request.
Start time newer than end time	GetEvents	The specified start time is later than the specified end time.
System busy; snapshot file not available	GetEvents	The OIS application is already performing an event log snapshot in response to a request from another client or operator.
System or file error	GetEvents	A memory allocation, system communication, or file error occurred in the processing of the request.
System or file error during snapshot request	GetEvents	A memory allocation, system communication, or file error occurred in the processing of the snapshot request.
The CIU is OFFLINE	AAG_GetList, AAG_GetTable, AAG_PutList, AAG_PutTable, AAG_Monitor, AAG_GetHistory, AAG_GetHistEvent, AAG_GetHistStat	The computer interface unit is off-line so no data is available.
Time-out (n) invalid	RequestDisplay	The time-out input value could not be converted to an integer or it is less than zero.
too few or too many parameters	GetEvents	The number of input values does not conform to the requirements of the command.
Unable to get log messages from file	Diagnose	An error occurred attempting to read the server diagnostic log file.
Unable to open log file	Diagnose	An error occurred attempting to open the server diagnostic log file.

Table 5-3. Error Messages (continued)

Return Error Text	Applicable Client Functions	Description
Unknown command %s	AAG_Command	The command string specified is not one of the following command strings recognized by the Elsag Bailey @aGlance/IT server: DiagnosticLog GetEvents GetWindow ListClients RequestDisplay SetOptions
Unrecognized input parameter <i>n</i>	GetEvents	The <i>n</i> th input value does not conform with the requirements of the command.
Window # <i>n</i> inactive	GetWindow, RequestDisplay	The console or workstation window is not active.

APPENDIX A - TAG ATTRIBUTE NAMES

INTRODUCTION

This section lists OIS 40 Series and Conductor VMS tag attribute names and provides general information on tag attribute names. The tag attribute names listed in this section provide a client platform access to Elsag Bailey systems using a DDE/OLE interface such as @aGlance/IT.

NOTE: Some of these tag attributes are specific to Harmony I/O.

DATA TYPES

Tag attributes have one of the following data types.

FLOAT	Double floating point.
STRING	Null terminated character string.
INTEGER	Longword integer.
BOOLEAN	Single character, an unsigned byte.
NODATA	Returned by a server that could not process a particular tag attribute pair.

TAG ATTRIBUTE CHARACTERISTICS

Attributes that have both a numeric and string representation return the text version by default. A suffix of _N is required to get the numeric representation. The specified string lengths are maximum string lengths and do not include the null terminator. The strings are not blank padded unless otherwise noted. The strings and string lengths for the attributes with a text representation are based on the strings currently used by the console and workstation. The strings may be changed by the operator through the text substitution feature to support foreign languages. The string lengths cannot be changed. The attribute names themselves are not substitutable.

TAG ATTRIBUTE NAMES

Tables A-1 through A-17 list tag attribute names for the most commonly used tag attributes. Tables A-18 through A-33 list tag attribute names for less used tag attributes. The less used tag attributes include attributes such as:

- Attributes used to derive other attributes. For, example, the alarm comments and alarm priorities for each alarm for

each tag type are used to derive the current alarm comment and priority.

- Attributes used internally by the console. For example, the flags indicating whether to log events and actions for each tag to the console or workstation event log.

Primary Tag Attributes

Tables A-1 through A-17 list primary tag attributes.

Table A-1. Common Attributes, All Tag Types

Attribute	Description	R/W/S ¹	Data Type	Value
C_DESC	Tag Description	R	String 32	Tag Description
EB_ALMSTS	Current alarm status	R/W/S ²	Integer String 2	0 blank no alarm 1 A Digital, N90STA or TEXSTR alarm 2 * Bad quality 3 N Return to normal 4 H High alarm 5 L Low alarm 6 HD High deviation alarm 7 LD Low deviation alarm 8 2H Two high alarm 9 2L Two low alarm 10 3H Three high alarm 11 3L Three low alarm 12 HR High rate alarm 13 LR Low rate alarm 14 HF Hardware alarm 15 Not used 16 !! Alarm suppressed
EB_BADQUAL	Good/Bad quality	R/W	Boolean String 4	0 = GOOD 1 = BAD
EB_BLOCK	Tag hardware block address	R	Integer	0 - 9998
EB_CURCMT	Alarm comment for current alarm state	R	String 64	User configured string
EB_CUPRIO	Alarm priority for current alarm state	R	Integer String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_CUSTTAGID	Customer tag ID	R	String 32	String from tag database
EB_ICIINDEX	ICI index number	R	Integer	1 - 30000
EB_LOOP	Tag hardware loop address	R	Integer	0 - 250
EB_MANINHB	Alarm manually inhibited flag	R/W	Boolean	0 = no manual inhibit 1 = manual inhibit
EB_MODULE	Tag hardware module address	R	Integer	0 -31

Table A-1. Common Attributes, All Tag Types (continued)

Attribute	Description	R/W/S¹	Data Type	Value
EB_NODE	Tag hardware node address	R	Integer	0 - 250
EB_PRDSP	Primary display name	R	String 8	Blank padded display name
EB_PRDPCRT	CRT on which to callup primary display	R	Integer	0 - 8, 0 is the current display
EB_QUAL	Quality	R/W	Integer String 11	0 blank Good 1 * Bad 2 x Off scan 3 s Substituted 4 ? Suspect 5 i Inhibit
EB_SCANSTS	Scan status	R/W	Boolean	0 = off scan 1 = on scan
EB_SUBST	Value is substituted value	R/W	Boolean	0 = not substituted 1 = substituted
EB_TAG	Tagname	R	String 14	User configured string
EB_TAGTYPE	Tag type	R	Integer String 8	0 = UNDEF 1 = ANALOG 2 = DIGITAL 3 = N90STA 4 = RCM 5 = STATION 6 = RMSC 7 = DD 8 = MSDD 9 = RMCB 10 = DAANALG 11 = DADIGTL 12 = TEXT 13 = DANG 14 = DADIG 15 = TEXTSTR 16 = INTANG 17 = INTDIG 18 = DEVSTAT
EB_XRTIMESTAMP	Exception report time stamp	R	String 20 or String 23	DD-MMM-YY HH:MM:SS or DD-MMM-YY HH:MM:SS.XX if the module time stamp feature is enabled

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. Can write to INTANG (non, H, L) and INTDIG (none, A) tag types. Can substitute TEXTSTR (none, A) tag types.
3. Can write to analog report (GOOD, BAD), digital report (GOOD, BAD), internal digital (GOOD, BAD), and internal analog (GOOD, BAD) tag types.
4. Can write to analog report, digital report, internal digital, internal analog, and text string tag types.

Table A-2. ANALOG Tag Attributes

Attribute	Description	R/W/S¹	Data Type	Value
C_APV	Analog value	R/S	Float	Expressed in engineering units
C_PVHIAP	High alarm limit value	R	Float	Expressed in engineering units
C_PVLOAP	Low alarm limit value	R	Float	Expressed in engineering units
C_RANGLO	Process variable zero scale	R	Float	Expressed in engineering units

Table A-2. ANALOG Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
C_UNITS	Engineering units descriptor	R	Integer String 8	All blanks if not available. 0 - 255 = Index into table of engineering descriptors. The string returned is the descriptor in the table corresponding to the index.
EB_CALIBSTS	Calibration status	R	Boolean String 14	0 = NORMAL 1 = OUT OF RANGE
EB_CONFIGERR	Configuration error exists flag.	R	Boolean String 14	0 = CONFIG OK 1 = CONFIG ERROR
EB_DISABLED	Point disabled	R	Boolean String 14	0 = NORMAL 1 = SERVICED
EB_IOTYPE	Distinguishes between standard and enhanced I/O types	R	Integer String 7	0 = STD 1 = ENH/IN 2 = ENH/OUT
EB_OVR	Override enable flag.	R/W	Boolean String 14	0 = NORM 1 = OVR
EB_OVRPERM	Override permissive flag.	R	Boolean String 14	0 = OVR PERMITTED 1 = OVR LOCKED OUT
EB_OVRVAL	Override value.	R/W	Float	float value
EB_PROPQUAL	Propagated quality.	R	Boolean String 14	0 = GOOD QUAL PROP ² 1 = BAD QUAL PROP ³
EB_RANGE	Range. Block I/O channel condition.	R	Integer String 14	0 = OK 1 = OPEN 2 = SHORT 3 = OVERDRIVE/LOW 4 = OVERDRIVE/HIGH
EB_READBACK	Readback Status.	R	Boolean String 14	0 = READBK STAT OK 1 = READBK STA BAD
EB_REFSTS	Reference Status.	R	Boolean String 14	0 = REF STATUS OK 1 = REF STATUS BAD
EB_SIMMODE	Simulation mode flag.	R	Boolean String 14	0 = NORMAL MODE 1 = SIM MODE
EB_SPAN	Scale span	R	Float	Expressed in engineering units
EB_STSERRINH	Status errors inhibited flag.	R	Boolean String 14	0 = NORMAL 1 = STAT ERR INHIB

NOTES:

1. Indicates the operations that can be performed on the tag read (R), write (W), and substitute (S).
2. GOOD QUAL PROP is only meaningful for the enhanced I/O types. This value means the propagated quality received is good.
3. BAD QUAL PROP is only meaningful for the enhanced I/O types. This value means the propagated quality received is bad.

Table A-3. DAANALG Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_ALMDB	Alarm deadband	R	Float	Expressed in engineering units
C_APV	Analog value	R/S	Float	Expressed in engineering units
C_PVHIAP	High alarm limit value	R	Float	Expressed in engineering units
C_PVLOAP	Low alarm limit value	R	Float	Expressed in engineering units
C_RANGLO	Process variable zero scale	R	Float	Expressed in engineering units

Table A-3. DAANALG Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
C_UNITS	Engineering units descriptor	R	Integer String 8	All blanks if not available. 0 - 255 = Index into table of engineering descriptors. The string returned is the descriptor in the table corresponding to the index
EB_HI2DELTA	High high alarm limit value	R	Float	Expressed in engineering units
EB_HI3DELTA	High high high alarm limit value	R	Float	Expressed in engineering units
EB_HIVARALMTAG	Variable high alarm limit tag	R	String 14	Blank padded tag name
EB_LO2DELTA	Low low alarm limit value	R	Float	Expressed in engineering units
EB_LO3DELTA	Low low low alarm limit value	R	Float	Expressed in engineering units
EB_LOVARALMTAG	Variable low alarm limit tag	R	String 14	Blank padded tag name
EB_SPAN	Scale span	R	Float	Expressed in engineering units

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-4. DADIG Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_DPV	Digital state	R/S	Boolean	0 = zero state 1 = one state The value returned or substituted is determined by EB_SRC.
C_SPV	Logic state descriptor of C_DPV	R	String 16	IF C_DPV is: 0 then string = contents of EB_LS0 1 then string = contents of EB_LS1
EB_ALMSUP	Enable/disable alarm suppression	W	Boolean	0 = unsuppress alarms 1 = suppress alarms
EB_CALCST	Calculated (alternate) state ²	R	Boolean	0 = zero state 1 = one state
EB_FORCEXR	Force an exception report	W	Boolean	0 = not used 1 = force an exception report
EB_LATCHED	Value/status latched flag	R/W	Boolean, String 14	0 = UNLATCHED 1 = LATCHED
EB_LS0	Zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_LS1	One state descriptor	R	String 16	Logic state descriptor for one state
EB_NOREPORT	Module level no report flag	R/W	Boolean, String 14	0 = ON REPORT 1 = NO REPORT
EB_PERINSEL	Permit input source selection flag	R	Boolean, String 14	0 = LOCKED 1 = UNLOCKED
EB_QUALOVR	Quality overridden flag	R	Boolean, String 14	0 = NORMAL 1 = OVERRIDDEN
EB_REALM	Re-alarm toggle	R	Boolean, String 14	0 = ZERO 1 = ONE
EB_REALST	Real (primary) state ³	W	Boolean	0 = zero state 1 = one state
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_SRC	Source of the reported value	R/W	Integer, String 14	0 = USER INSERTED 1 = ALTERNATE 2 = PRIMARY

Table A-4. DADIG Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_USERST	User inserted state ²	R/W	Boolean	0 = zero state 1 = one state
EB_OUTRANGE	Determines if the value is unconstrained and out of range	R	Boolean String 12	0 = IN RANGE 1 = OUT OF RANGE
EB_USERTYPE	User type code	R	Integer	0 - 255

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. This attribute causes the value to be read from the block. Use it to obtain the calculated value when the DADIG is not in alternate mode.
3. This attribute causes the value to be read from the block. Use it to obtain the real value when the DADIC is not in primary mode.

Table A-5. DAANG (DANG Tag Type) Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_APV	Digital state	R/W/S	Float	Expressed in engineering units
C_UNITS	Engineering units descriptor	R	Integer String 8	All blanks if not available. 0 - 255 = Index into table of engineering descriptors. The string returned is the descriptor in the table corresponding to the index
EB_ALMSUP	Enable/disable alarm suppression	W	Boolean	0 = disabled 1 = enabled
EB_CALCVAL	Calculated value	R	Float	Expressed in engineering units
EB_CNTRREF	Center display reference value	R	Float	Expressed in engineering units
EB_CONSTR	Value constrained flag	R	Boolean, String 14	0 = UNCONSTRAINED 1 = CONSTRAINED
EB_CONSTRPER	Constraint limits enabled flag	R	Boolean, String 14	0 = DISABLED 1 = ENABLED
EB_FORCEXR	Force an exception report	W	Boolean	1 = force exception report
EB_HICONSTRLIMIT	High constraint limit value	R	Float	Expressed in engineering units
EB_HIREF	High display reference value	R	Float	Expressed in engineering units
EB_LOCONSTRLIMIT	Low constraint limit value	R	Float	Expressed in engineering units
EB_LOREF	Low display reference value	R	Float	Expressed in engineering units
EB_MULTLVL	Multi-level alarming enabled flag	R	Boolean, String 14	0 = DISABLED 1 = ENABLED
EB_NOREPORT	Module level no report flag	R/W	Boolean, String 14	0 = ON REPORT 1 = NO REPORT
EB_NXTHILMT	Next higher limit	R	Float	Expressed in engineering units
EB_NXTLOLMT	Next lower limit	R	Float	Expressed in engineering units
EB_PERINSEL	Permit input source selection flag	R	Boolean, String 14	0 = LOCKED 1 = UNLOCKED
EB_QUALOVR	Quality overridden flag	R	Boolean, String 14	0 = NORMAL 1 = OVERRIDDEN
EB_REALM	Re-alarm toggle	R	Boolean, String 14	0 = ZERO 1 = ONE
EB_REALVAL	Real state	W	Boolean	Expressed in engineering units
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_SRC	Source of the reported value	R/W	Integer, String 14	0 = USER INSERTED 1 = ALTERNATE 2 = PRIMARY
EB_USERVAL	User inserted state	R/W	Float	Expressed in engineering unit

Table A-5. DAANG (DANG Tag Type) Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_OUTRANGE	High or low constraint limits exceeded status	R	Boolean String 14	1 = OUT OF RANGE 0 = IN RANGE
EB_HARDFAULT	Hardware fault status	R	Boolean String 14	1 = HARDWARE FAIL 0 = OK
EB_VARALM	Variable alarms enabled flag	R	Boolean String 14	0 = DISABLED 1 = ENABLED

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-6. DADIGTL/INTDIG Tag Attributes

Attribute ¹	Description	R/W/S ²	Data Type	Value
C_DPV	Digital state	R/W/S	Boolean	0 = zero state 1 = one state
C_SPV	Symbolic state	R	String 16	IF C_DPV is: 0 then string = contents of EB_LS0 1 then string = contents of EB_LS1
EB_ALMDEF	Alarm state definition	R	Integer, String 8	0 = ZERO 1 = ONE 2 = NEITHER
EB_LS0	Zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_LS1	One state descriptor	R	String 16	Logic state descriptor for one state

NOTES:

1. Only C_DPV and C_SPV apply to the OIS 40 Series console or Conductor VMS workstation for the INTDIG attribute.

2. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-7. Device Driver (DD) Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_DPV	Digital state	R/W/S ²	Boolean	0 = reset control output 1 = set control output
C_SPV	Symbolic state	R/W ³	String 16	IF C_DPV is: 0 then string = contents of EB_LS0 1 then string = contents of EB_LS1
EB_DDMODE	Device driver mode	R/W ⁴	Integer String 6	0 = AUTO 1 = REMOTE 2 = MANUAL
EB_FB1	Feedback 1 state	R	Boolean, String 16	0 = EB_FB1LS0 1 = EB_FB1LS0
EB_FB1LS0	Feedback 1 zero state descriptor	R	String 16	Logic state descriptor for feedback 1 zero state
EB_FB1LS1	Feedback 1 one state descriptor	R	String 16	Logic state descriptor for feedback 1 one state
EB_FB2	Feedback 2 state	R	Boolean, String 16	0 = EB_FB2LS0 1 = EB_FB2LS0
EB_FB2LS0	Feedback 2 zero state descriptor	R	String 16	Logic state descriptor for feedback 2 zero state
EB_FB2LS1	Feedback 2 one state descriptor	R	String 16	Logic state descriptor for feedback 2 one state
EB_FBSTS	Feedback status	R	Boolean, String 14	0 = NORMAL 1 = BAD
EB_LS0	Zero state descriptor	R	String 16	Logic state descriptor for C_DPV zero state

Table A-7. Device Driver (DD) Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_LS1	One state descriptor	R	String 16	Logic state descriptor for C_DPV one state
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_STSOVR	Status override flag	R	Boolean, String 14	0 = NORMAL 1 = OVERRIDDEN

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. Can only write in manual mode.
3. OIS 40 Series console and Conductor VMS workstation cannot write.
4. Can only write auto and manual.

Table A-8. Digital Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_DPV	Digital state	R/S	Boolean	0 = zero state 1 = one state
C_SPV	Symbolic state	R	String 16	IF C_DPV is: 0 then string = contents of EB_LS0 1 then string = contents of EB_LS1
EB_ALMDEF	Alarm state definition	R	Integer, String 8	0 = ZERO 1 = ONE 2 = NEITHER
EB_CONFIGERR	Configuration error exists flag.	R	Boolean String 14	0 = CONFIG OK 1 = CONFIG ERROR
EB_IOTYPE	Distinguishes between standard and enhanced I/O types.	R	Boolean String 14	0 = STD 1 = ENH/IN 2 = ENH/OUT
EB_LS0	Zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_LS1	One state descriptor	R	String 16	Logic state descriptor for one state
EB_OVR	Override enable flag.	R/W	Boolean String 6	0 = NORM 1 = OVR
EB_OVRPERM	Override permissive flag.	R	Boolean String 14	0 = OVR PERMITTED 1 = OVR LOCKED OUT
EB_OVRST	Override state.	R/W	Boolean String 6	0 = <zero logic state descriptor (EB_LS0)> 1 = <one logic state descriptor (EB_LS1)>.
EB_PROPQUAL	Propagated quality.	R	Boolean String 14	0 = GOOD QUAL PROP ² 1 = BAD QUAL PROP ³
EB_RANGE	Range. Block I/O channel condition.	R	Integer String 14	0 = OK 1 = OPEN 2 = SHORT 3 = OVERDRIVE
EB_READBACK	Readback Status.	R	Boolean String 14	0 = READBK STAT OK 1 = READBK STA BAD
EB_SIMMODE	Simulation mode flag.	R	Boolean String 14	0 = NORMAL MODE 1 = SIM MODE
EB_STSERRINH	Status errors inhibited flag.	R	Boolean String 14	0 = NORMAL 1 = STAT ERR INHIB

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. GOOD QUAL PROP is only meaningful for the enhanced I/O types. This value means the propagated quality received is good.
3. BAD QUAL PROP is only meaningful for the enhanced I/O types. This value means the propagated quality received is bad.

Table A-9. INTANG Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_APV	Analog value	R/W/S	Float	Expressed in engineering units
C_PVHIAP	High alarm limit value	R	Float	Expressed in engineering units
C_PVLOAP	Low alarm limit value	R	Float	Expressed in engineering units
C_RANGLO	Process variable zero scale	R	Float	Expressed in engineering units
C_UNITS	Engineering units descriptor	R	Integer String 8	All blanks if not available. 0 - 255 = Index into table of engineering descriptors. The string returned is the descriptor in the table corresponding to the index.
EB_CALIBSTS	Calibration status	R	Boolean String 14	0 = NORMAL 1 = OUT OF RANGE
EB_DISABLED	Point disabled	R	Boolean String 14	0 = NORMAL 1 = SERVICED
EB_SPAN	Scale span	R	Float	Expressed in engineering units

NOTES:

1. Indicates the operations that can be performed on the tag read (R), write (W), and substitute (S).

Table A-10. Multistate Device Driver (MSDD) Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_DPV	Digital state of control output 1	R	Boolean	0 = Control output = zero 1 = Control output = one
C_IPV	Last good state	R/S ²	Integer	0 - 3
C_ISP	Requested state	R/W ³	Integer	0 - 3 Can only write 1 - 3.
C_SPV	Symbolic last good state	R	String 16	If C_IPV is: 0 then string = contents of EB_LS0 1 then string = contents of EB_LS1 2 then string = contents of EB_LS2 3 then string = contents of EB_LS3
EB_AUTOMAN	Auto manual mode	R/W	Boolean, String 6	0 = MANUAL 1 = AUTO
EB_CNTRLOVR	Control override flag	R	Integer String 14	0 = NORMAL 1 = OVERRIDDEN
EB_FB1	Feedback 1 state	R	Boolean, String 16	0 = EB_FB1LS0 1 = EB_FB1LS1
EB_FB1LS0	Feedback 1 zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_FB1LS1	Feedback 1 one state descriptor	R	String 16	Logic state descriptor for one state
EB_FB2	Feedback 2 state	R	Boolean, String 16	0 = EB_FB2LS0 1 = EB_FB2LS1
EB_FB2LS0	Feedback 2 zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_FB2LS1	Feedback 2 one state descriptor	R	String 16	Logic state descriptor for one state
EB_FB3	Feedback 3 state	R	Boolean, String 16	0 = EB_FB3LS0 1 = EB_FB3LS1
EB_FB3LS0	Feedback 3 zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_FB3LS1	Feedback 4 one state descriptor	R	String 16	Logic state descriptor for one state
EB_FB4	Feedback 4 state	R	Boolean, String 16	0 = EB_FB4LS0 1 = EB_FB4LS1
EB_FB4LS0	Feedback 4 zero state descriptor	R	String 16	Logic state descriptor for zero state

TAG ATTRIBUTE NAMES
Table A-10. Multistate Device Driver (MSDD) Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_FB4LS1	Feedback 4 one state descriptor	R	String 16	Logic state descriptor for one state
EB_FBSTS	Feedback status	R	Boolean, String 14	0 = NORMAL 1 = BAD
EB_LS0	Zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_LS1	One state descriptor	R	String 16	Logic state descriptor for one state
EB_LS2	Two state descriptor	R	String 16	Logic state descriptor for zero state
EB_LS3	Three state descriptor	R	String 16	Logic state descriptor for one state
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_STSOVR	Status override flag	R	Boolean, String 14	0 = NORMAL 1 = OVERRIDDEN

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. Can only substitute 1 through 3.
3. Can only write 1 through 3.

Table A-11. N90STA Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_BACCFG	Redundancy configured	R	Boolean	0 = no redundancy 1 = redundancy
EB_BACSTS	Backup status flag	R	Boolean, String 14	0 = GOOD 1 = BAD
EB_BACTAG	Tag name for redundant module	R	String 14	Blank padded tag name
EB_LOCIOSTS	Local I/O error flag	R	Boolean, String 14	0 = nor error 1 = error
EB_MODCFGERR	Module configuration error flag	R	Boolean	0 = no error 1 = error
EB_MODERR	Module errors exist flag	R	Boolean	0 = no error 1 = error
EB_MODMODE	Module mode	R	Integer, String 6	0 = CONFIGURE 1 = FAILED 2 = ERROR 3 = EXECUTE 4 = STANDBY ²
EB_MODMODEERR	Module mode error flag	R	Boolean	0 = error 1 = no error
EB_MODTYP	Module type	R	Integer, String 8	1 = PIM 2 = CTM
EB_NCFGERR	Node configuration error flag	R	Boolean	0 = error 1 = no error
EB_NCOMMERR	Node communication error flag	R	Boolean	0 = error 1 = no error
EB_NMODERR	Module errors in a node exist flag	R	Boolean	0 = error 1 = no error
EB_NOFFLINE	Off-line flag	R	Boolean	0 = on line 1 = off line
EB_NSYSERR	Node system status error flag	R	Boolean	0 = error 1 = no error
EB_REMIOSTS	Remote I/O status	R	Boolean, String 14	0 = NO ERROR 1 = ERROR

Table A-11. N90STA Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_SECONDARY	Primary/secondary flag	R	Boolean	0 = primary mode 1 = secondary mode

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. Standby status is equivalent to a module in execute mode and secondary in a redundant pair of modules.

Table A-12. Remote Control Memory (RCM) Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_DPV	Digital state	R/W/S	Boolean	0 = zero state 1 = one state
C_SPV	Symbolic state	R	String 16	If C_DPV is: 0 then string = contents of EB_LS0 1 then string = contents of EB_LS1
EB_FEEDBACK	Feedback state indicating the actual state of the controlled device	R	Boolean, String 16	0 = EB_LS0 1 = EB_LS1
EB_LS0	Zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_LS1	One state descriptor	R	String 16	Logic state descriptor for one state
EB_OUTOVR	Output override flag	R	Boolean, String 14	0 = NOT SET 1 = SET
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_RESCMD	Reset command	R	Boolean, String14	0 = NOT SET 1 = SET
EB_RESIN	Reset input value	R	Boolean, String14	0 = NOT RESET 1 = RESET
EB_SETCMD	Set command	R	Boolean, String14	0 = NOT SET 1 = SET
EB_SETIN	Set input value	R	Boolean, String14	0 = NOT SET 1 = SET
EB_SETPER	Set permissive flag	R	Boolean, String14	0 = NOT SET 1 = SET

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-13. RMCB Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_DPV	Digital state of control output 1	R/W/S	Boolean	0 = zero state 1 = one state
C_SPV	Symbolic last good state	R/W ²	String 16	If C_DPV is: 0 then string = contents of EB_LS0 1 then string = contents of EB_LS1
EB_BADSTART	Bad start flag	R	Boolean, String 14	0 = GOOD 1 = BAD
EB_ERRCODE	Error code	R	Integer, String 20	0 = NO ERROR 1 = STOPPED 2 = INTERLOCK 1 LOW 3 = INTERLOCK 2 LOW 4 = INTERLOCK 3 LOW 5 = INTERLOCK 4 LOW 6 = FEEDBACK 1 LOW

Table A-13. RMCB Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ERRCODE (continued)	Error code	R	Integer, String 20	7 = FEEDBACK 2 LOW 8 = FEEDBACK 1 HIGH 9 = FEEDBACK 2 HIGH
EB_FAULT	Fault flag	R	Boolean, String 14	0 = NO FAULT 1 = FAULT
EB_FB1	Feedback 1 state	R	Boolean, String 16	0 = EB_FB1LS0 1 = EB_FB1LS1
EB_FB1LS0	Feedback 1 zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_FB1LS1	Feedback 1 one state descriptor	R	String 16	Logic state descriptor for one state
EB_FB2	Feedback 2 state	R	Boolean, String 16	0 = EB_FB2LS0 1 = EB_FB2LS1
EB_FB2LS0	Feedback 2 zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_FB2LS1	Feedback 2 one state descriptor	R	String 16	Logic state descriptor for one state
EB_LS0	Zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_LS1	One state descriptor	R	String 16	Logic state descriptor for one state
EB_PM1LS0	Permissive 1 zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_PM1LS1	Permissive 1 one state descriptor	R	String 16	Logic state descriptor for one state
EB_PM2LS0	Permissive 2 zero state descriptor	R	String 16	Logic state descriptor for zero state
EB_PM2LS1	Permissive 2 one state descriptor	R	String 16	Logic state descriptor for one state
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_STARTPM1	Start permissive 1 state	R	Boolean String 16	0 = EB_PM1LS0 1 = EB_PM1LS1
EB_STARTPM2	Start permissive 2 state	R	Boolean String 16	0 = EB_PM2LS0 1 = EB_PM2LS1

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. Only read (no write) for OIS 40 Series console and Conductor VMS workstation.

Table A-14. RMSC Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_APV	Analog value/set point	R/W/S	Float	Expressed in engineering units
EB_OPHILM	High limit value	R	Float	Expressed in engineering units
EB_OPLOLM	Low limit value	R	Float	Expressed in engineering units
C_RANGLO	Process variable zero scale	R	Float	Expressed in engineering units
C_UNITS	Engineering units descriptor	R	Integer String 8	All blanks if not available. 0 - 255 = Index into table of engineering descriptors. The string returned is the descriptor in the table corresponding to the index
EB_TRACKING	Output tracking flag	R	Boolean, String 14	0 = NO TRACK 1 = TRACKING
EB_SPAN	Scale span	R	Float	Expressed in engineering units

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-15. Station Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
C_AOUTPUT EB_VAL3	Control output (CO)	R/W/S	Float	Expressed in engineering units
C_APV EB_VAL1	Analog process variable (PV)	R/S	Float	Expressed in engineering units
C_ASP EB_VAL2	Analog set point (SP)	R/W/S	Float	Expressed in engineering units
C_PVHIAP	High alarm limit value	R	Float	Expressed in engineering units
C_PVLOAP	Low alarm limit value	R	Float	Expressed in engineering units
C_RANGLO	Process variable zero scale	R	Float	Expressed in engineering units
C_RATIO	Ratio index value (RI)	R/W/S	Float	0.0 if station is not in ratio mode
C_UNITS	Engineering units descriptor	R	Integer String 8	All blanks if not available. 0 - 255 = Index into table of engineering descriptors. The string returned is the descriptor in the table corresponding to the index
EB_AUTOMAN	Auto/manual mode flag	R/W	Boolean, String 6	0 = MANUAL 1 = AUTO
EB_BYPASS	Bypass flag	R	Boolean, String 14	0 = NO BYPASS 1 = BYPASS
EB_CMPTRSTS	Computer OK status flag	R/W	Boolean, String 14	0 = FAILED 1 = GOOD
EB_COMMANDOUT ²	Command output value/mode	W	Integer, String 32	0 = LOCAL MANUAL 1 = LOCAL AUTO 2 = LOCAL CASCADE 3 = COMPUTER MANUAL 4 = COMPUTER AUTO 5 = COMPUTER CASCADE 6 = LOCAL LEVEL 7 = COMPUTER LEVEL 8 = COMPUTER BACKUP 9 = COMPUTER OK 10 = PREVIOUS STATE
EB_DEVLMT	Deviation alarm limit value	R	Float	Expressed in engineering units
EB_MODELOCK	Manual mode interlock flag	R	Boolean, String 14	0 = NO MODE LOCK 1 = MODE LOCK
EB_OUTSTS	Station analog output status	R	Boolean, String 14	0 = GOOD 1 = BADE
EB_OUTTRACKING	Output tracking flag	R	Boolean, String 14	0 = NO TRACK 1 = TRACKING
EB_SPAN	Process variable span	R	Float	Expressed in engineering units
EB_SPSPAN	Setpoint span	R	Float	Expressed in engineering units
EB_PVSTS	Process variable bad quality flag	R	Boolean, String 14	0 = GOOD 1 = BAD
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_SPTRACKING	Setpoint tracking enabled flag	R	Boolean, String 14	0 = NO TRACK 1 = TRACKING
EB_SPZERO	Setpoint zero scale	R	Float	Expressed in engineering units
EB_STNVL	Station level	R/W	Boolean, String 14	0 = LOCAL 1 = COMPUTER
EB_STNMODE	Cascade/ratio or normal flag	R/W	Boolean, String 14	0 = NORMAL 1 = CASCADE/RATIO

Table A-15. Station Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_PVQUAL	PV quality	R	Boolean String 14	0 = GOOD 1 = BAD
EB_STNTYPE	Station type	R	Integer, String 32	0 = BASIC WITH SET POINT 1 = BASIC WITHOUT SET POINT 2 = BASIC WITH BIAS 3 = RATIO 4 = CASCADE

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. Not supported by the OIS 40 Series console or Conductor VMS workstation. Instead, use the following attributes: EB_AUTOMAN, EB_STNLVL, EB_STNMODE, EB_CMPTRSTS.

Table A-16. TEXT Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
EB_BLINK	Text blink enabled flag	R/S	Boolean	0 = no blink 1 = blink
EB_COLOR	Text color	R/S	Integer, String 12	0 = BLACK 1 = WHITE 2 = RED 3 = GREEN 4 = BLUE 5 = CYAN 6 = MAGENTA 7 = YELLOW 8 = ORANGE 9 = YELLOW GREEN 10 = GREEN CYAN 11 = CYAN BLUE 12 = BLUE MAGENTA 13 = MAGENTA RED 14 = DARK GRAY 15 = LIGHT GRAY
EB_MSG	Message number	R/S	Integer, String 80	0 - 9999 A configured text string stored in the supported system is associated with each message number.

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-17. TEXTSTR Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMSUP	Enable/disable alarm suppression	R/W	Boolean	0 = disabled 1 = enabled
EB_AUTOMAN	Auto/manual mode	R/W	Boolean, String 6	0 = MANUAL 1 = AUTO
EB_BLKLEN	Module defined text length	R	Integer	1 - 80
EB_CNTRLPER	Control permissible flag	R	Boolean, String 14	0 = DISABLED 1 = ENABLED
EB_MODELOCK	Mode interlock	R	Boolean, String 14	0 = NO MODE LOCK 1 = MODE LOCK
EB_OISLEN	Console text length	R	Integer	1 - 80
EB_PDTECHO	Echo flag	R	Boolean, String 14	Exception report from block Exception report from control

Table A-17. TEXTSTR Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_QUALOVR	Quality overridden flag	R/W ²	Boolean, String 14	0 = NORMAL 1 = OVERRIDDEN
EB_RCVLEN	Actual length	R	Integer	Up to 80 characters
EB_REALM	Re-alarm toggle	R	Boolean, String 14	0 = ZERO 1 = ONE
EB_REDTAG	Red tagged	R	Boolean, String 14	0 = NORMAL 1 = RED TAGGED
EB_SEQNUM	Sequence number of this exception report	R	Integer	0 - 255
EB_STRLOCK	String interlock enabled flag	R	Boolean, String 14	0 = UNLOCKED 1 = LOCKED
EB_TRUNC	Text string truncation flag	R	Boolean, String 14	0 = NO TRUNCATION 1 = REMOTE TRUNCATION
EB_TXTSTR	Text string data	R/W/S	String 80	Up to an 80 character text string ³

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. Read only (no write) for OIS 40 Series console and Conductor VMS workstation.
3. For OIS 40 Series console and Conductor VMS workstation, may be limited by EB_OISLEN.

Secondary Tag Attributes

Table A-18 through A-33 list secondary tag attributes.

Table A-18. All Tag Type Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
C_ACK	Digital state	R	Boolean	0 = no 1 = acknowledged
C_ADPKB	Symbolic state	R	Integer	0 - 8, where 0 = all keyboards
EB_ADPLAMP	Alarm state definition	R	Integer	0 - 32, where 0 = none
EB_ADANEL	Zero state descriptor	R	Integer	0 - 4, where 0 = none
EB_ALMGRP	Alarm group	R	Integer, String 2	0 - 99 100 = S 101 = D
EB_AUTOINHB	Alarm auto inhibited flag	R	Boolean	0 = no 1 = alarm auto inhibited
EB_BADPRIO	Alarm priority for bad quality alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_GBLACK	Transmit global acknowledge flag	R	Boolean	1 = global acknowledge
EB_GRPINHB	Inhibited by alarm group flag	R	Boolean	1 = transmit global acknowledge
EB_INHBONST	Inhibit on state flag	R	Boolean	True for digital state reporting tags only

Table A-18. All Tag Type Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_INHBST	Alarm inhibiting alarm status or digital state	R	Integer, String 2	For EB_INHBONST = true: 0 - 3 For EB_INHBONST = false: H, 2H, 3H, L, 2L, 3L, HD, LD, HR, LR, A, or blank ²
EB_INHBTAG	Alarm inhibit tag	R	String 14	Blank padded tag name or all blanks (disabled) ³
EB_MANINHBPER	Manual inhibiting of alarms enabled flag	R	Boolean	1 = manual inhibit of alarms enabled
EB_NODELIST	Node list number for global alarm management	R	Integer	0 = node list built by the console at start-up from the first 32 N90 STA tags 1 - 4 = configured node list
EB_PRTALM	Print alarm messages flag	R	Boolean	1 = print alarm messages
EB_PRTOACT	Print operator action messages flag	R	Boolean	1 = print operator messages
EB_RTN	Return to normal flag	R	Boolean	1 = return to normal
EB_RTNCMT	Alarm comment for return to normal state	R	String 64	1 - 64 characters
EB_RTNPRI	Alarm priority for return to normal alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_SAVALM	Save alarm messages flag	R	Boolean	1 = save alarm messages
EB_SAVOACT	Save operator action messages flag	R	Boolean	1 = save operator action messages
EB_SECGRP	Security group	R	Integer	1 - 16
EB_SECLVL	Security level	R	Integer	1 - 16
EB_SPECRCVD	Exception report specifications have been received flag	R	Boolean	1 = exception report specifications received

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).
2. The valid values also depend on the tag type of the alarm inhibit tag.
3. Cannot be an RMSC tag.

Table A-19. Analog/INTANG Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_DECDIG	Floating point digits	R	Integer	0 -4
EB_HICMT	Alarm comment for high alarm	R	String 64	0 - 64 characters
EB_HIPRI	Alarm priority for high alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM

Table A-19. Analog/INTANG Tag Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_LOCMT	Alarm comment for low alarm flag	R	String 64	0 - 64 characters
EB_LOPRIO	Alarm priority for low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-20. DAANALG Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMTYPE	Alarm type	R	Integer, String 8	STANDARD: high to low FIXED: 3 high to 3 low VARIABLE: dynamic high and low
EB_DECDIG	Symbolic state	R	Integer	0 - 4
EB_HI2CMT	Alarm comment for high high alarm	R	String 64	0 - 64 characters
EB_HI2PRIO	Alarm priority of high high alarm	R	Integer String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_HI3CMT	Alarm comment for high high high alarm	R	String 64	0 - 64 characters
EB_HI2PRIO	Alarm priority of high high high alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_HICMT	Alarm comment for high alarm	R	String 64	0 - 64 characters
EB_HIPRIO	Alarm priority of high alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LO2CMT	Alarm comment for low low alarm	R	String 64	0 - 64 characters

Table A-20. DAANALG Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_LO2PRIO	Alarm priority of low low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LO3CMT	Alarm comment for low low low alarm	R	String 64	0 - 64 characters
EB_LO3PRIO	Alarm priority of low low low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LOCMT	Alarm comment for low alarm	R	String 64	0 - 64 characters
EB_LOPRIO	Alarm priority of low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-21. DAANG Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_DECDIG	Floating point digits	R	Integer	0 - 4
EB_HI2CMT	Alarm comment for high high alarm	R	String 64	0 - 64 characters
EB_HI2PRIO	Alarm priority of high high alarm	R	Integer	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_HI3CMT	Alarm comment for high high high alarm	R	String 64	0 - 64 characters

Table A-21. DAANG Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_HI2PRIO	Alarm priority of high high high alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_HICMT	Alarm comment for high alarm	R	String 64	0 - 64 characters
EB_HIPRIO	Alarm priority of high alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_HIDEVCMT	Alarm comment for high deviation alarm	R	String 64	0 - 64 characters
EB_HIDEVPRIO	Alarm priority of high deviation alarm	R	Integer String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_HIRATECMT	Alarm comment for high rate alarm	R	String 64	0 - 64 characters
EB_HIRATEPRIO	Alarm priority of high rate alarm	R	Integer String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LO2CMT	Alarm comment for low low alarm	R	String 64	0 - 64 characters
EB_LO2PRIO	Alarm priority of low low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LO3CMT	Alarm comment for low low low alarm	R	String 64	0 - 64 characters

Table A-21. DAANG Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_LO3PRIO	Alarm priority of low low low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LOCMT	Alarm comment for low alarm	R	String 64	0 - 64 characters
EB_LOPRIO	Alarm priority of low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LODEVCMT	Alarm comment for low deviation alarm	R	String 64	0 - 64 characters
EB_LODEVPRIO	Alarm priority of low deviation alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LORATECMT	Alarm comment for low rate alarm	R	String 64	0 - 64 characters
EB_LORATEPRIO	Alarm priority of low rate alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-22. DADIG Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for digital alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for digital alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_PRTSTCHG	Print state change messages flag	R	Boolean	1 = print state change messages

Table A-22. DADIG Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_SAVSTCHG	Save state change message flag	R	Boolean	1 = save state change messages

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-23. Device Driver (DD) Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for digital alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for digital alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_PRTSTCHG	Print state change messages flag	R	Boolean	1 = print state changes
EB_SAVSTCHG	Alarm priority of high high alarm	R	Boolean	1 = save state changes

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-24. DEVSTAT Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for DEVSTAT ² alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for DEVSTAT alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_DEVTYPE	Device type	R	Integer, String 16	0 = undefined 1 = printer
EB_PRTNUM	Printer number	R	Integer	1 - 4
EB_PRTTYPE	Printer type	R	Integer, String 10	0 = ANSI 1 = ANSI color 2 = IBM 3 = IBM color 5 = ANSI high 6 = IBM B&W 24 7 = IBM col24

NOTES:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

2. Only applicable on the OIS 40 Series console and Conductor VMS workstation for determining the printer type. The device status for a printer is not maintained via the device status tag.

Table A-25. Digital/DADIGTL/INTDIG Tag Attributes

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for digital alarm	R	String 64	0 - 64 characters

Table A-25. Digital/DADIGTL/INTDIG Tag Attributes (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMPRIO	Alarm priority for digital alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_PRTSTCHG	Print state change messages flag	R	Boolean	1 = print state change messages
EB_SAVSTCHG	Save state change message flag	R	Boolean	1 = save state change messages

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-26. MSDD Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for digital alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for digital alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_PRTSTCHG	Print state change messages flag	R	Boolean	1 = print state change messages
EB_SAVSTCHG	Alarm priority of high high alarm save state change message flag	R	Boolean	1 = save state change messages

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-27. N90STA Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for N90STA alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for N90STA alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_MODCFGACK	Module configuration error acknowledged flag	R	Boolean	1 = module configuration error acknowledged
EB_MODERRACK	Module errors exist acknowledged flag	R	Boolean	1 = module errors exist acknowledged
EB_MODMODEACK	Module mode error acknowledged flag	R	Boolean	1 = module mode error acknowledged
EB_MODRTN	Communications module return to normal flag	R	Boolean	1 = communications module return to normal

Table A-27. N90STA Tag Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_NCFGACK	Node configuration error acknowledged flag	R	Boolean	1 = node configuration error acknowledged
EB_NCOMMACK	Node communication error acknowledged flag	R	Boolean	1 = node communication error acknowledged
EB_NMODERRACK	Module errors in a node exist acknowledged flag	R	Boolean	1 = module errors in a node exist acknowledged
EB_NOFFLINEACK	Node off-line error acknowledged flag	R	Boolean	1 = Node off-line error acknowledged
EB_NSYSACK	Node system status error acknowledged flag	R	Boolean	1 = node system status error acknowledged
EB_PCUINHB	PCU inhibit flag	R	Boolean	1 = PCU inhibit
EB_PCUSCANSTS	PCU scan off flag	R	Boolean	1 = PCU scan off
EB_XMITACK	Transmit global acknowledge enabled flag	R	Boolean	1 = transmit global acknowledge enabled
EB_XMITSIL	Transmit global silence enabled flag	R	Boolean	1 = transmit global silence enabled

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-28. RCM Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for digital alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for digital alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_PRTSTCHG	Print state change messages flag	R	Boolean	1 = print state change messages
EB_SAVSTCHG	Alarm priority of high high alarm save state change message flag	R	Boolean	1 = save state change messages

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-29. RMCB Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for digital alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for digital alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_PRTSTCHG	Print state change messages flag	R	Boolean	1 = print state change messages

TAG ATTRIBUTE NAMES

Table A-29. RMCB Tag Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_SAVSTCHG	Alarm priority of high high alarm save state change message flag	R	Boolean	1 = save state change messages
EB_TXTSET	Text set number	R	Integer	0 -100, 0 indicates no text set

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-30. RMSC Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_DECDIG	Floating point digits	R	Integer	0-4

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-31. Station Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_DECDIG	Floating point digits	R	Integer	0 - 4
EB_HICMT	Alarm comment for high alarm	R	String 64	0 - 64 characters
EB_HIDEVCMT	Alarm comment for high deviation alarm	R	String 64	0 - 64 characters
EB_HIDEVPRIO	Alarm priority of high deviation alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_HIPRIO	Alarm priority for high alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_LOCMT	Alarm comment for low alarm	R	String 64	0 - 64 characters
EB_LODEVCMT	Alarm comment for low deviation alarm	R	String 64	0 - 64 characters
EB_LODEVPRIO	Alarm priority for low deviation alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM

Table A-31. Station Tag Attributes (Secondary) (continued)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_LOPRIO	Alarm priority for low alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM
EB_TUNEBLK	Block number of PID used for tuning	R	Integer	11 - 9998

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-32. TEXT Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_PRTSTCHG	Print state change messages flag	R	Boolean	1 = print state change messages
EB_SAVSTCHG	Alarm priority of high high alarm save state change message flag	R	Boolean	1 = save state change messages

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Table A-33. TEXTSTR Tag Attributes (Secondary)

Attribute	Description	R/W/S ¹	Data Type	Value
EB_ALMCMT	Alarm comment for TEXTSTR alarm	R	String 64	0 - 64 characters
EB_ALMPRIO	Alarm priority for TEXTSTR alarm	R	Integer, String 8	1 - 8, 1 has the highest priority 1 = DIRE 2 = CRUCIAL 3 = CRITICAL 4 = ALERT 5 = WARNING 6 = ADVISE 7 = NOTIFY 8 = INFORM

NOTE:

1. Indicates the operations that can be performed on the tag, read (R), write (W), and substitute (S).

Index

A

AAG.INI file 4-24
AAG_Admin utility 3-4
AAG_Cancel Monitor 4-2
AAG_CheckPermission 2-3
AAG_Client 4-2
AAG_ClientInit 2-3, 4-1
AAG_ClientInit() 4-1
AAG_ClientInt 4-4
AAG_ClientTerminate 2-3, 4-2, 4-4
AAG_Command 2-4, 4-1, 4-3, 5-1
AAG_Dispatch 4-4
AAG_EndSession 4-3
AAG_Function 4-6
AAG_GetAttrs 4-3, 4-9
AAG_GetCurrentSession 4-33
AAG_GetErrorText 4-3, 5-7
AAG_GetHistEvent 4-3, 4-20
AAG_GetHistory 4-3, 4-19
AAG_GetHistStat 4-3
AAG_GetList 4-3, 4-9
AAG_GetTable 4-4, 4-9
AAG_GetTags 4-4
AAG_HIST_AVG 4-3
AAG_HIST_MAX 4-3
AAG_HIST_MIN 4-3
AAG_INSERT 4-4
aag_ListServers 3-4
AAG_Monitor 4-4, 4-14
AAG_MonitorInit 4-4
AAG_MonitorTerminate 4-4
AAG_NoData 4-20
AAG_PutHistory 4-4
AAG_PutList 4-4
AAG_PutTable 4-5
AAG_REPLACE 4-4
AAG_Shutdown 4-5
AAG_StartSession 4-5
AAG_StopServer 3-4
AAG_StopSession 3-4
AAG_Success 4-9
AAG_ValueList 4-8
Activation command 4-33
ADDHOST 3-2, 3-4
AllAttrs 4-7
Alpha AXP/Open VMS 1-2
ANALOG/INTANG A-3, A-9
ANALOG/INTANG Attributes A-16
Attribute
 Characteristics A-1
Attributes 4-10, A-1

ANALOG/INTANG A-3, A-9, A-16
Common A-15
DAANALG A-4, A-17
DAANG A-18
DADIG A-5, A-20
Device Driver (DD) A-7
DEVSTAT A-21
Digital/DADIGTL/INTDIG A-21
Digital/INTDIG/DADIG A-7, A-8
Hidden 4-7
MSDD A-9, A-22
N90STA A-10, A-22
Primary A-1
RCM A-11, A-23
RMCB A-11
RMSB A-23
RMSC A-12, A-24
Secondary 4-9
Station A-13, A-24
TEXT A-14, A-25
TEXTSTR A-14, A-25

B

Boolean data type A-1

C

Callback
 Event_Notify 4-35
Client
 @aGlance/IT 2-2
 Platforms 2-1
 Software installation 3-2
ClientNumber 4-6
Command
 ClientNumber 4-6
Commands
 Diagnose 5-1
 Diagnostic Log 4-32
 GetEvents 2-4
 GetList 4-7
 GetWindow 4-30
 Imhostid 3-5
 ListClient 4-6
 RequestDisplay 4-31
 SetOptions 4-7
Common Attributes A-2
Common attributes A-2
Configuration 2-5
Console level write attributes 4-10

Index (continued)

D

DAANALG Attributes.....	A-4
DAANG	A-18
Attributes	A-6
DAANG Attributes	A-6
DADIG	A-5, A-20
Data monitoring.....	2-4
Data types	A-1
DDE/OLE	2-1
Device Driver (DD).....	A-21
Device Driver (DD) Attribute.....	A-7
DEVSTAT.....	A-21
Diagnose.....	5-1
Command options	5-1
Diagnostic log.....	4-32
Digital/DADIGTL/INTDIG	A-21
Digital/INTDIG/DADIG.....	A-7, A-8
Documentation.....	1-5

E

EBOIS-WRITE permission.....	3-4
end_time	4-25
Error messages.....	5-7
Event log data	2-4, 4-24
Event_Notify.....	4-35
Exception monitoring.....	4-4, 4-8

F

flexlm.dat.....	3-6
Floating data type	A-1
Function	
AAG_ClientTerminate	4-2
AAG_GetHistory.....	4-19
Function key.....	4-37
Functions	
AAG_CancelMonitor.....	4-2
AAG_CheckPermissions	2-3
AAG_Client.....	4-2
AAG_ClientFlnt	4-4
AAG_ClientInit	2-3, 4-1
AAG_ClientTerminate	2-3, 4-4
AAG_Command	2-4, 4-1, 4-3, 5-1
AAG_Dispatch.....	4-4
AAG_EndSession.....	4-3
AAG_Function	4-6
AAG_GetAttrs.....	4-3
AAG_GetCurrentSession	4-33
AAG_GetErrorText.....	4-3, 5-7
AAG_GetHistEvent.....	4-20
AAG_GetHistory.....	4-3

AAG_GetHistStat.....	4-3
AAG_GetList.....	4-3
AAG_GetTable	4-4
AAG_GetTags	4-4
AAG_HIST_AVG	4-3
AAG_HIST_MAX	4-3
AAG_HIST_MIN	4-3
AAG_INSERT	4-4
AAG_Monitor	4-4, 4-14
AAG_MonitorInit	4-4
AAG_MonitorTerminate	4-4
AAG_PutHistory.....	4-4
AAG_PutList	4-4
AAG_PutTable.....	4-5
AAG_REPLACE	4-4
AAG_Shutdown	4-5
AAG_StartSession.....	4-5
GetAttr	4-7
GetHistEvent.....	4-3

G

GetEvents.....	2-4
GetList	4-7
GetTable	4-7
GetWindow	4-30

H

Hidden attributes	4-7
Historical data.....	2-4, 4-17
HOSTID	3-5
How to use manual.....	1-4

I

Imhostid command	3-5
Input	
AllAttrs	4-7
Installation	3-1
Integer data type.....	A-1

K

Key select.....	4-35
-----------------	------

L

License key.....	3-1
ListClient.....	4-6

M

- Manual content 1-4
- module level write operations 4-10
- Monitoring data 2-4
- Monitoring data points 4-4
- MSDD A-22
- Multistate Device Driver (MSDD)..... A-9

N

- N90STA Attributes A-10, A-22
- Network configuration 3-2
 - @aGlance/IT 3-3
 - @aGlance/IT clients 3-4
- Next index 4-24
- No Data 4-9
- No data return A-1
- Nomenclature 1-6
 - semAPI 1-6

O

- Operating systems 1-2

P

- Parameters
 - Key select 4-36
 - Touch point 4-35
- Permissions 2-3
 - AAG_ListServers 3-4
 - AAG_StopSession 3-4
 - EBOIS-WRITE 3-4
 - StopServer 3-4
- Polled monitoring 4-4, 4-7
- Primary attributes 4-7, A-1
- Product features 1-3
- Proxies 2-3

R

- RCM A-23
- RCMB A-11
- Real time data 2-4
- Real time tag data 4-9
- Reference time retrieval 4-28
- Remote Control Memory (RCM)..... A-11
- Returns
 - No Data 4-9
- RMSB A-23
- RMSC A-24
- RMSC Attributes A-12

S

- Secondary attributes 4-7, 4-9
- semAPI nomenclature 1-6
- Server
 - Configuration 2-5
 - INFI 90 OPEN 2-1
 - Platforms 2-1
 - Software license 3-1
- Server platforms 1-2
- Server status 5-4
- SetOptions 4-7
- SetOptions command 4-7
- SHOWHOSTS 3-3
- Software
 - License key 3-1
 - Registration 3-5
- Software
 - Loading 3-5
- start_index 4-25
- start_time 4-25
- Station A-24
- Station attributes A-13
- Status messages 5-4
- String data type A-1
- SYSTEM account 3-4
- System requirements 1-2

T

- Tag attribute names A-1
- Tagname access 2-3
- TCP/IP network 2-1
- TEXT A-25
- Text attributes A-14
- TEXTSTR A-25
- TEXTSTR Attributes A-14
- Time span retrieval 4-28
- Touch point 4-34
- Trend types 4-17

U

- User information 1-1
- User task definition 4-37
- User task interface 4-1
- UTI Functions 4-1

W

- Writable attributes 4-10

Visit Elsag Bailey on the World Wide Web at <http://www.ebpa.com>

Our worldwide staff of professionals is ready to meet *your* needs for process automation.
For the location nearest you, please contact the appropriate regional office.

AMERICAS

29801 Euclid Avenue
Wickliffe, Ohio USA 44092
Telephone 1-216-585-8500
Telefax 1-216-585-8756

ASIA/PACIFIC

152 Beach Road
Gateway East #20-04
Singapore 189721
Telephone 65-391-0800
Telefax 65-292-9011

EUROPE, AFRICA, MIDDLE EAST

Via Puccini 2
16154 Genoa, Italy
Telephone 39-10-6582-943
Telefax 39-10-6582-941

GERMANY

Graefstrasse 97
D-60487 Frankfurt Main
Germany
Telephone 49-69-799-0
Telefax 49-69-799-2406

Form WBPEEU1350255B0 Litho in U.S.A. 0797

Copyright © 1997 by Elsag Bailey Process Automation, As An Unpublished Work

® Registered Trademark of Elsag Bailey Process Automation

™ Trademark of Elsag Bailey Process Automation

