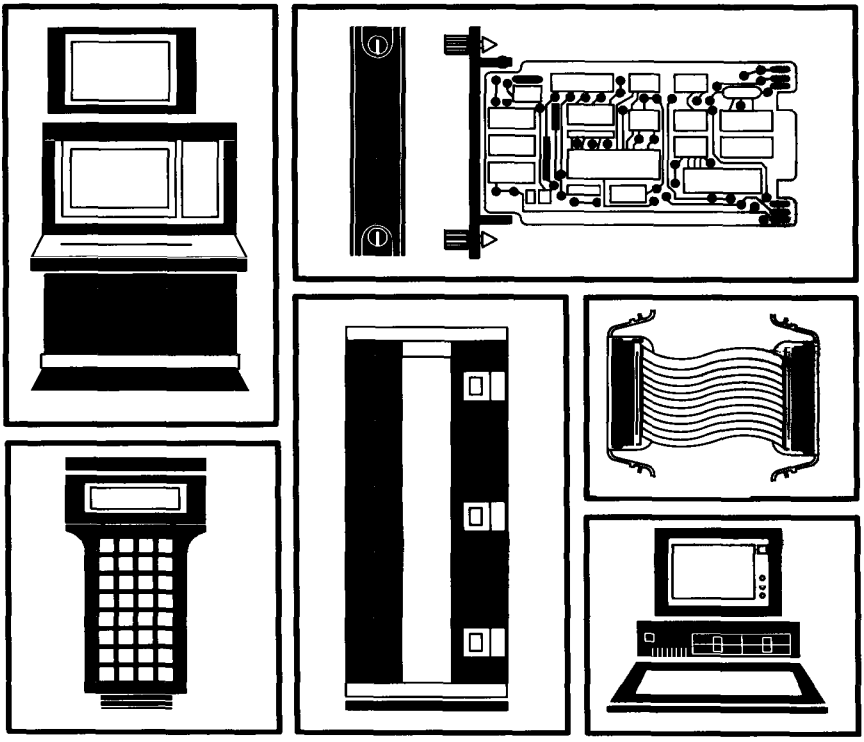
Bailey infi 90

ODS/ODC

# Instruction

## Open Data Server/Client



***WARNING*** not ces as used n th s manua app y to hazards or unsafe pract ces wh ch cou d resu t n personal njury or death

***CAUTION*** not ces app y to hazards or unsafe pract ces wh ch cou d resu t n property damage

## WARNING

***Instruction Manuals***

Do not nsta ma nta n or operate th s equ pment w thout read ng, understand ng and fo ow ng the proper ABB nstruct ons and manua s otherw se njury or dam age may resu t

***Radio Frequency Interference***

Most e ectron c equ pment s nf uenced by rad o frequency nterference (RF ) Caut on shou d be exercised w th regard to the use of portab e commun cat ons equ pment n the area around such equ pment Prudent pract ce d ctates that s gns shou d be posted n the v c n ty of the equ pment caut on ng aga nst the use of portab e commun cat ons equ pment

***Possible Process Upsets***

Ma ntenance must be performed on y by qua f ed personne and on y after secur ng equ pment contro ed by th s product Adjust ng or remov ng th s product wh e t s n the system may upset the process be ng contro ed Some process upsets may cause njury or damage

## NOTICE

The nformat on conta ned n th s document s subject to change w thout not ce

ABB, ts aff ates emp oyees, and agents and the authors of and contr butors to th s pub cat on spec f ca y d sclaim all l abilities and warranties express and implied (including warranties of merchantability and fitness for a part cu ar purpose) tor the accuracy currency, comp eteness and /or re ab ty of the nformat on conta ned here n and/or for the f tness for any part cu ar use and/or for the performance of any mater a and/or equ pment se ected n who e or part w th the user of/or n re ance upon nformat on conta ned here n Se ect on of mater a s and/or equ pment s at the sole r sk of the user of th s pub cat on

Th s document conta ns propr etary nformat on of ABB and s ssued n str ct conf dence ts use or reproduct on for use, for the reverse eng neer ng deve opment or manufacture of hardware or software descr bed here n s proh b ted No part of th s document may be photocop ed or reproduced w thout the pr or wr tten consent of ABB

# 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 Link ing 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 devel oped 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 diag nostic 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 viii | 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

# Table of Contents (continued)



# List of Figures

# List of Tables

# List of Tables (cont nued)

# 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 Elsag 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 Elsag 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 Link ing 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 devel oped 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 appli cation 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 vari ous third party tools such as relational databases, expert sys tems, 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 soft ware 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 corpo rate personnel should have a working knowledge of server/cli ent operation on the applicable platform Use of the third party application software is beneficial Experience operating Sym phony and Symphony and INFI 90 OPEN systems and applica tions 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

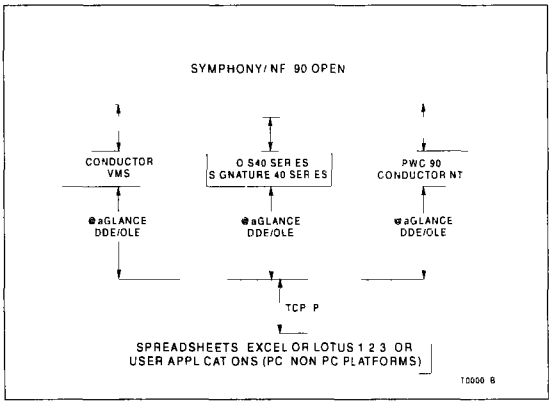



*Figure 1 1 @aGlance/IT Application Overview*

- 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 |
|---|---|
| E 96 192 2 | Configuration ( S42 and IIO S42) |
| 1 E96 197 | Display Builder Reference ( S42 and O S42) |
| WBPEEUI220023B0 | File Utilities ( S42 and IIO S42) |
| WBPEEU 220756A0 | OIS40 Services Operation |
| WBPEEU 220757A0 | O S40 Services Configuration |
| WBPEEU 220758A0 | O S40 Services File Utilities |
| WBPEEU 220759A0 | O S40 Services Display Builder |
| WBPEEU 220762A0 | Configuration Persona Work Center 90 |
| WBPEEUI220773A0 | Operation Conductor VMS |
| WBPEEU 220774A0 | Configuration Conductor VMS |
| WBPEEUI220775A0 | File Utilities Conductor VMS |

*Table 1 2 Reference Documents (continued)*

| Number | Title |
|---|---|
| WBPEEU 220776A0 | Display Bu der Reference |
| WBPEEU 220779A0 | Configuration Conductor NT |
| WBPCEL 900014A0 | @aG ance/ T User's Gu de consist ng of<br>@aGlance/IT for M crosoft Windows<br>@aG ance/ T for M crosoft Exce<br>@aG ance/IT for Lotus 1 2 3 |
| WBPCEU 900020A0 | @aG ance/ T nsta lat on and Operat on |

## 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**<br>@aG ance/ T |
| | | | | | | | 1 | | | **Platform**<br>O S 40 ser es<br>Signature 40 series and<br>Conductor VMS |
| | | | | | | | | 1 | | **Simultaneous Clients**<br>5 (1 server) |
| | | | | | | | | 2 | | 10 (2 servers) |
| | | | | | | | | | 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**<br>@aG ance/ T |
| | | | | | | | 1 | | | **Operating System**<br>MS W ndows (3 1 3 11<br>95, NT) |
| | | | | | | | | 1 | | **Media**<br>3 5 in 1 44 Mbyte f oppy |
| | | | | | | | | | 0 | Reserved |

## GLOSSARY OF TERMS AND ABBREVIATIONS

Table 1 5 contains definitions of terms and abbreviations that are unique to Elsag Bailey

*Table 1 5 Glossary*

| Term | Definition |
|---|---|
| Estab sh | A mechan sm wh ch enab es except on reports to flow between system nodes |
| Node | A po nt of nterconnect on to a network |
| Node Address | A un que dent fier of a spec f c dev ce or a communicat on channel Refers to P ant Loop Super oop or NFI NET address |
| O S | Operator nterface stat on ntegrated operator conso e w th data acqu sition and report ng capabil ties t prov des a digita access into the process for flexible control and mon tor ng |

# 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 sys tems, 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 c ent p atforms may be supported as they are devel oped and re eased by Intu tive 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 cli ents, 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 con trol 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

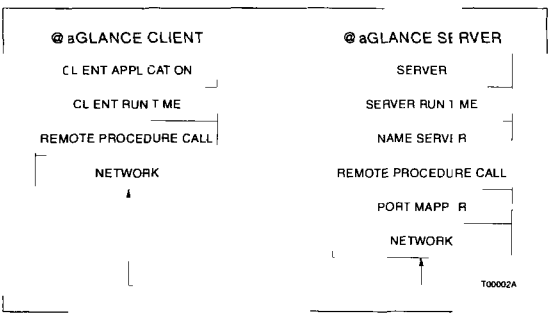



*Figure 2 1 @aGlance/IT Server/Client Architecture*

### @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 privi leges 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 con trol privileges, it cannot write data to the control system or server The @aGlance/IT provides both interpolated and com pressed 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 com mand 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 termi nates 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 com monly 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 sup ports 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 cli ent 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 pro cedure 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 unli censed 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 con nected 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 proce dure 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 pro cedure is also required when adding licensed servers capa ble of serving simultaneous clients

  **NOTE:** Th s procedure s also found n the appropr ate ***File Utilities*** manua It does not need to be repeated f a ready performed

- ***@aGLANCE/IT SERVER CONFIGURATION (WRITE ACCESS).*** These instructions explain on how to assign per missions to clients for write access to a server Complete the steps in this section to assign the **EBOIS-WRITE** per mission to clients

- ***ADDING ADDITIONAL @aGLANCE/IT CLIENTS*** Addi tional 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/ T can be established the console has to be configured on the TCP/ P network with the PORTMAPPER feature enabled and the TCP/ P 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* Return

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* Return

where

*client name* @aGlance/IT client name

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

**NOTES:**
1 This command also updates the HIO C42 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** Return

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** Return

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** Return

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 cli ents 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 capabil ity to write to a server, is also available The **EBOIS-WRITE** per mission 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 con figuration 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 man ual listed in Table 1 2 for system reset procedures

Refer to the **@aGlance/IT Installation and Operations** man ual for a discussion on proxy accounts, permission assign ments 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* Return

### @aGlance LICENSE

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

### 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 Elsag Bailey There are two methods available for software registration phone registration through Elsag Bailey Customer Services Center or registration via a computer with a modem through the Elsag Bailey Process Automation License Administration System The following items are needed to register the @aGlance/IT software

- The *HOSTID* which is generated using the **lmhostid** com mand Refer to Step 1 under ***License Installation***
- The Elsag Bailey Software Registration Form which con tains 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 **lmhostid** 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** Enter

5 Enter the appropriate information at the prompt and press Enter

*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 automati cally 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 |
| ut destroy xdspnam | Not supported |
| uti request display | AAG Command |
| ut diagnostic log | AAG Command |
| uti strerror | AAG GetErrorText |
| ut _print_text f le | Not supported |
| uti define reports | Not supported |
| ut fcp mode | Not supported |
| uti output values | Not supported |

*Table 4 1 UTI/@aGlance Functions (continued)*

| UTI Function | @aGlance API Function |
|---|---|
| ut connect | AAG StartSession |
| ut disconnect | AAG EndSession |
| ut get tag indices | Not supported |
| uti output stn value | AAG PutList, AAG PutTable |
| ut set stn values | AAG PutList AAG PutTable |
| ut read values | AAG GetList AAG GetTable |
| ut write int values | AAG PutList, AAG PutTable |
| ut read tag data | AAG GetList |
| ut get osi time | Not supported |
| ut set oasis time | Not supported |
| ut convert time | Not supported |
| ut_get trend | AAG GetHistory |
| ut 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 GetAttrs** 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 *dd-mmm yyyy hh mm ss*

**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 *dd mmm yyyy hh mm ss* and the interval time is *ss* (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 *dd mmm yyyy hh mm ss* 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 Elsag Bailey @aGlance/IT servers

**AAG GetTags** This function returns tag names known to the indicated server It is supported by all Elsag 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 parame ters such as status list, monitor type, interval, monitor argu ments, argument count, client data, and monitor ID The optional parameter monitor type indicates the type of monitor ing 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 his torical 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 sup ported 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 ses sions in progress. Some servers that have not registered the corresponding callback function will always refuse this request A client application requires the **AAG StopServer** per mission 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 It the client does not user this function, then a session is automatically started on the specified server when one of the other @aGlance/IT cli ent functions is invoked

*Table 4 2 Supported @Glance/IT API Functions*

| Function Type | API Function | OIS 40 Series/ Conductor VMS Server Support |
|---|---|---|
| Error handling | AAG GetErrorText | • |
| Sess on control | AAG_StartSession | • |
| | AAG EndSess on | • |
| | AAG Shutdown | • |
| Real t me data - read | AAG GetLIst | • |
| | AAG GetTable | • |
| | AAG Monitor | • |
| | AAG NextMonitor | • |
| | AAG MonitorArgs | • |
| Real t me data write | AAG PutLIst | • |
| | AAG PutTable | • |
| H stor ca data -read | AAG GetHistEvent | • |
| | AAG GetHistory | • |
| | AAG GetHistStat | • |
| Historical data - wr te | AAG PutH story | Not supported |
| Tag data and attr butes | AAG GetTags | • |
| | AAG GetAttrs | • |
| User task activat on | AAG ClientIn t | • |
| | AAG ClientTerm nate | • |
| 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 CancelMon tor | • |
| | AAG Monitor | • |
| | AAG_Mon tor n t | • |
| | 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 |
|---|---|---|---|---|
| EBO S | Bart | O S | 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 | O S | 1 000 | 29 MAY 1996 08 07 30 |
| **Client Node** | **User Name** | **Connect Time** | **Last Activity Time** | **Function Count** |
| NODE1 | | 29-MAY · 996 14 50 34 | 29-MAY 1996 14 57 20 | 20 |
| | | | | |

| Server Name | Node | Type | Version | Start Time |
|---|---|---|---|---|
| EBO S | Bart | O S | 1 000 | 29 MAY 1996 08 07 30 |
| ***Client Node*** | ***User Name*** | ***Connect Time*** | ***Last Activity Time*** | ***Function Count*** |
| Act vity Counts | | | | |
| GetTags | 2 | GetAttrs4 | GetL st 0 | |
| GetTab e | 4 | PutL st0 | PutTab e0 | |
| Monitor | 0 | Mon torCance 0 | Mon torUpdate9 | |

**NOTE.** n Excel the output va ue ce is shou d be set to a f xed w dth font such as Courier so that the columns a gn

### *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 on y sets the server funct ona ty w th respect to the current c ient

*Table 4 3 Set Options Arguments*

| Option | Description |
|---|---|
| A lAttrs | Enab es the server to return h dden attr butes |
| NoA Attrs | Disables the server from returning hidden attr butes |
| PolledMonitoring ( nt) | Spec f es that the server should hand e mon toring requests from th s cl ent where the monitor type s not exp icit y set in the mon tor request as be ng pol ed type Where (int) s the po l ng interva in sec onds A va ue must be entered for ( nt), otherw se no polling w occur |

*Table 4 3 Set Options Arguments (continued)*

| Option | Description |
|---|---|
| NoPol edMonitor ng | Specif es that the server shou d hand e mon tor ng requests from th s c ent where the mon tor type s 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** **server* | Name of the server |
| **char** **command* | Server specific command name, **"SetOptions"** |
| **AAG ValueList** *in values* | Options listed in Table 4 3 |
| **int** *in count* | Count of input values in the in values list |
| **AAG ValueList** *out values* | Not used |
| **int** *out count* | 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,**
**Po ledMonitoring=***int***,1'**

Where

| | |
|---|---|
| *YouıServer* | name of the server whose monitoring behavior is being modified |
| *int* | 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 s requ red (as shown n the example DDE Adv se message)

2 The last digit of the DDE Adv se message command s a vert ca ne not the letter l

The **SetOptions** command must be invoked from the same cli ent that is used for monitoring Monitoring from Excel must be done using DDE Advise messages These messages communi cate via DDE with the local @aGlance/IT DDE Bridge applica tion, 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 sec ondary attributes, use the **AAG Command** server specific **Set-Options** 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** func-tions This option is enabled (set to yes) via the general param eters 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 work station 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 rec ommended that a status list be used in all calls that support them

If the **Display Last Good Value** option is enabled, then the qual ity attribute (EBG QUAL) should always be requested with the tag value Refer to ***ALARM QUALITY OPTIONS*** in the appropri ate 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 Attributes[1], Console Level*

| Tag Type | Attribute | Description | Data Type | Values[5] |
|---|---|---|---|---|
| ANALOG/ NTANG | EB MANINHB | A arm manua ly nh bited f ag | Boolean | 0 = not manua nhib t<br>1 = manua nhi> t |
| | EB SCANSTS[3] | Scan status | Boo ean | 0 = off scan<br>1 = on scan |
| | EB SUBST[2] | Value s subst tuted value | Boo ean | 0 – not subst tuted<br>1 – substituted |
| DAANLAG DAANG RMSC | C APV | Analcg va ue | Float | Expressed n er g neer ng un ts |
| | EB MANINHB | Alarm manua y inh bited flag | Boo ean | 0 = not manua nh b t<br>1 – manual inh bit |
| | EB SCANSTS[3] | Scan status | Boolean | 0 – off scan<br>1 = on scan |
| | EB SUBST[2] | Va ue s subst tuted va ue | Boo ean | 0 – not substituted<br>1 – substituted |
| DAD G DADIGTL, DD, DIG TAL RCM RMCB | C DPV | Dig ta state | Boo ean | 0 = zero state<br>1 = one state |
| | EB MAN NHB | A arm manua ly nh bited flag | Boolean | 0 – not manua nh b t<br>1 = manua nh bit |
| | EB SCANSTS[3] | Scan status | Boo ean | 0 off scan<br>1 on scan |
| | EB SUBST[2] | Value is subst tuted va ue | Boolean | 0 not substituted<br>1 substituted |
| DEVSTAT | EB MANINHB | Alarm manually inhibited flag | Boolean | 0 not manual nhibit<br>1 manua nhi> t |
| | EB SCANSTS[3] | Scan status | Boolean | 0 off scan<br>1 on scan |
| MSDD | C PV | Last good state | Integer | 0 3 |
| | EB MAN NHB | Alarm manual y inhibited flag | Boo ean | 0 – not manua nhibit<br>1 manua nhi> t |
| | EB SCANSTS[3] | Scan status | Boolean | 0 off scan<br>1 on scan |
| | EB SUBST[2] | Value is subst tuted value | Boo ean | 0 not subst tu ed<br>1 subst tuted |

*Table 4 4 Writable Attributes[1], Console Level (continued)*

| Tag Type | Attribute | Description | Data Type | Values[5] |
|---|---|---|---|---|
| STAT ON[4] | C APV | Analog va ue | F oat | Expressed n eng neer ng un ts |
| | C AOUTPUT | Contro output (CO) | F oat | Expressed as % of span |
| | C ASP | Analog set point (SP) | F oat | Expressed n eng neer ng un ts |
| | C RATIO | Rat o ndex va ue (R ) | Float | 0 0 if stat on s not n ratio mode |
| | I B MAN NHB | A arm manual y nh bited f ag | Boo ean | 0 not manua nh b t<br>1 manua nh b t |
| | I B SCANSTS[3] | Scan status | Boo ean | 0 off scan<br>1 on scan |
| | I B SUBST[2] | Va ue s subst tuted value | Boo ean | 0 not subst tuted<br>1 subst tuted |
| TEXT | EB BL NK | *Text bl nk enab ed flag* | *Boolean* | *0 – no b nk*<br>1 b nk |
| | B COLOR | Text co or | Integer | 0 black<br>1 white<br>2 – red<br>3 green<br>4 b ue<br>5 – cyan<br>6 magenta<br>7 yel ow<br>8 orange<br>9 ye ow green<br>10 – green cyan<br>11 = cyan b ue<br>12 – blue magenta<br>13 magenta red<br>14 dark gray<br>15 – ight gray |
| | EB MSG | Message number | nteger | 1- 10000 |
| | EB MAN NHB | *Alarm manua ly inhib ted f ag* | Boo ean | 0 not ma ua nh b t<br>1 manua nh b t |
| | EB SCANSTS[3] | Scan status | Boo ean | 0 – off scan<br>1 on scan |
| | EB SUBST[2] | Va ue is subst tuted va ue | Boolean | 0 – not subst tuted<br>1 subst tuted |
| TEXTSTR | EB ALMSTS | Current alarm status | Integer | 0 no alarm<br>1 a arm |
| | EB TXTSTR | Text string data | Str ng 80 | Logic state descr ptor for one state |
| | EB MANINHB | Alarm manually nh b ted flag | Boo ean | 0 – not manual inh bit<br>1 manual inh b t |
| | EB SCANSTS[3] | Scan status | Boo ean | 0 – off scan<br>1 – on scan |
| | EB SUBST[2] | Va ue s subst tuted va ue | Boolean | 0 – not subst tuted<br>1 – substituted |

*Table 4 4 Writable Attributes[1]. Console Level (continued)*

| Tag Type | Attribute | Description | Data Type | Values[5] |
|---|---|---|---|---|
| NTANG | C APV | Analog va ue | F oat | Expressed n eng neering units |
| | EB ALMSTS | Current a arm status | Boo ean | 0 – no alarm<br>1 a arm |
| | EB QUAL | Quality | Boolean | 0 good<br>1 bad |
| | EB MANINHB | Alarm manua y inh b ted f ag | Boolean | 0 not manua inh bit<br>1 manual nh ɔ t |
| | EB SCANSTS[3] | Scan status | Boo ean | 0 – off scan<br>1 – on scan |
| NTD G | C DPV | Dig ta state | Boo ean | 0 zero state<br>1 – one state |
| | EB ALMSTS | Current alarm status | Boolean | 0 no alarm<br>1 a arm<br>4 high alarm<br>5 ow alarm |
| | EB QUAL | Qua ty | Boolean | 0 good<br>1 bad |
| | EB MANINHB | A arm manua y nhibited t ag | Boo ean | 0 not manual nh b t<br>1 – manua inhibit |
| | EB SCANSTS[3] | Scan status | Boo ean | 0 off scan<br>1 on scan |

**NOTES**

1 Some of the wnte operat ons are not nstantaneous espec a y the modu e eve wnte operat ons Therefore t may take awh e for the conso e to receive the changed va ue n an exception report

2 To place a tag into subst tution the EB SUBST attribute must be written with a value of one and a subst tutable attribute must immed ately fo ow t n the same request

3 When an on scan s ssued the c ent must verify the act on has completed before attempt ng another act on ( e con ro ) f contro s attempted mmediately after an on scan the contro va ues are assumed to be subst tuted va ues resu t ng n an a ready estab shed error

4 The STAT ON tag must be n computer eve before the mode or any va ue can be changed The STAT ON eve can a ways be set Wnt ng EB CMPTRSTS 1 and EB STNLVL 1 n a s ngle ca w work as long as EB CMPTRSTS appears f rst n the nput st

5 f an error occurs n wnt ng the va ue for a tag/attr bute pa r the correspond ng item n the status st f supp ed w conta n a message nd cat ng the fa ure The functions return **AAG Success** even f an error s encountered wnt ng the specific tag-attnbute pa rs t s strongly suggested that a status st be used n al ca s that support them

*Table 4 5 Writable[1] Attributes, Module Level*

| Tag Type | Attribute | Description | Data Type | Value |
|---|---|---|---|---|
| DADIG | EB ALMSUP | Enab e/disable alarm sup-press on | Boolean | 0 unsuppress alarms<br>1 suppress a arms |
| | EB FORCEXR | Force an exception report | Boolean | 0 – not used<br>1 force an exception report |
| | EB LATCHED | Va ue/status atched flag | Boo ean | 0 reset atch<br>1 not wr table |
| | EB NOREPORT | Modu e eve no report flag | Boo ean | 0 off report<br>1 on report |
| | EB SRC | Source of the reported value | Integer | 0 user nserted<br>1 = alternate<br>2 pr mary |
| | EB USERST | User nserted state | Boo ean | Eng neering un t |

*Table 4 5 Writable[1] Attributes, Module Level (continued)*

| Tag Type | Attribute | Description | Data Type | Value |
|---|---|---|---|---|
| DAANG | EB ALMSUP | Enab e/d sable alarm sup-pression | Boolean | 0 – unsuppress alarms<br>1 suppress alarms |
| | EB FORCEXR | Force an exception report | Boo ean | 0 – not used<br>1 force an except on report |
| | EB NOREPORT | Modu e level no report flag | Boo ean | 0 off report<br>1 – on report |
| | EB SRC | Source of the reported va'ue | nteger | 0 – user nserted<br>1 – a ternate<br>2 – primary |
| | EB USERVAL | User nserted state | F oat | Expressed in eng neering unit |
| DD | C DPV | D gital state | Boolean | 0 – zero state<br>1 = one state |
| | EB DDMODE | Device driver mode | Integer | 0 auto<br>1 = remote<br>2 – manual |
| MSDD | C_ISP | Requested state | Integer | 1 - 3 |
| | EB_AUTOMAN | Auto manual mode | Boolean | 0 – manual<br>1 – auto |
| RCM, RCMB | C DPV | D gital state | Boolean | 0 – zero state<br>1 one state |
| RMSC | C APV | Analog value | Float | Expressed in eng neering units |
| STATION | C ASP | Ana og set point (SP) | F oat | Expressed in engineering units |
| | C_AOUTPUT | Control output (CO) | Float | Expressed as % of span |
| | C_RATIO | Ratio ndex value (RI) | Float | 0 0 if station is not in ratio mode |
| | EB AUTOMAN | Auto manual mode | Boolean | 0 – manual<br>1 – auto |
| | EB_STNVL | Station level | Boolean | 0 – local<br>1 – computer |
| | EB STNMODE | Cascade/ratio or norma f ag | Boolean | 0 – norma<br>1 cascade/rat o |
| | EB CMPTRSTS | Computer OK status flag | Boolean | 0 – unused<br>1 = user flag |
| TEXTSTR | EB TXTSTR | Text string data | String 80 | Up to 80 characters text str ng |
| | EB ALMSUP | Enable/disable alarm sup-pression | Boolean | 0 = unsuppress alarms<br>1 suppress a arms |
| | EB ALMSUP | Enable/disable alarm sup press on | Boolean | 0 – unsuppress a arms<br>1 – suppress alarms |
| | EB AUTOMAN | Auto manual mode | Boo ean | 0 – manual<br>1 – auto |

**NOTE** 1 Some of the write operations are not nstantaneous especia ly the modu e eve write operations Therefore t may take awh e for the console to receive the changed va ue n 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** **server name* | Name of the server |
| **AAGNameList** *tag list* | Array of pointers to the tag names of the data points to be monitored |
| **AAGNameList** *attr list* | Array of pointers to attribute names this list must match the number of tag names |
| **int** *list count* | Number of tag/attribute pairs n the tag and attribute lists |
| **AAGValueList** *value list* | Array that contains the initial values for each tag/attribute pair |
| **AAGStatusList** *status list* | Optional parameter array that contains the initial status for each value |
| **char** **monitor type* | Optional parameter, indicates the type of monitoring to be done Use **AAG POLLED** for polled type monitoring, **AAG EXCEPTION** for exception type requests |
| **char** **interval* | Optional parameter, monitoring time interval (for polled type requests) String represents number of seconds in the interval (i e , 10 – 10 seconds) |
| **AAGValueList** *monitor args* | Optional parameter, client specific argument list |
| **int** *arg count* | Optional parameter, number of items in the monitor arguments list |
| **AAGULong** *clien' data* | Optional parameter, client specific data that can be associated with this monitor request |
| **AAGULong** *monitor id* | 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 set ting 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 cli ent 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**· Excess ve oad ng can occ r f a arge number of tag/ attr butes are mon tored in a arge number of monitor requests

If an on line tag configuration change is made to a tag cur rently being monitored, the configuration changes will not always be reflected in the monitored data If the tag configura tion 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 over all 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 pro viding 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 effi ciently 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 moni toring requests containing more tag/attribute pairs should be used where possible There is a notable performance improve ment 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 sus pension period expires in three seconds, after which updates will resume for that client provided no other monitor cancella tion 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 (func tion 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 resolu tion 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 possi ble to define two enhanced trends for a single tag mode sub type 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. Stan dard (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 (e ANALOG STATION DANG etc)
3 STATION tags only
4 Digital tag types (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** *tag list* | A list of character string pointers to the tag names |
| **AAGNameList** *attr list* | A list of character string pointers to the attribute names Each entry in the tag_list may have an attribute associated with it This list must be the same length as the tag_list C APVAVE is an example of an *attr list* 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 tag_list |
| **char** * *start time* | The start time is a string in the form *dd-mmm yyyy hh mm ss* |
| **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** *time list* | 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** *value table* | The server returns the historical values for the specified data points in this table |
| **AAGStatusList** *status table* | 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 num ber 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 his torical 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 com pressed 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 , com pressed 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** *tag list* | A list of requested tag names. |
| **AAGNameList** *attr list* | A list of requested attribute names C_APVAVE is an example of an *attr list* 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 *dd-mmm-yyyy hh mm ss* |
| **char** * *end time* | The end time is a string in the form *dd-mmm yyyy hh mm ss* |
| **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 time list, and the number of rows or samples in the value and status table |
| **AAGTimeList** *time list* | Returned by the server, a list of times specifying when a sample was recorded |
| **AAGValueList** *value table* | 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 consecu tively, 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** *status table* | The server returns a status value for each entry in value table 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** *time index* | 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** *tag list* | A list of requested tag names |

**AAGNameList** *attr list* — A list of requested attribute names C APVAVE is an example of an *attr list* 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 *dd mmm-yyyy hh mm ss*

**char** * *end time* — The end time is a string in the form *dd-mmm yyyy hh mm ss*

**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** *value table* — 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** *status table* — If this parameter is nonzero, the server should return a status for each entry in value table. 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 sys tem 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 con figuration 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 parame ter 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 com plete 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 particu lar 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** func tion 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** *in values* — A list of values to be sent to the server Table 4 7 lists the ordered input parameters that *in values* must contain These parameters determine the time span retrieval case

**int** *in count* — The number of input values (*in values*) being sent.

**AAGValueList** *out values* — A value list to store returning data from the server

**int** *out count* — The number of entries allocated in the output value list (*out values*)

*Table 4 7 Time Span Retrieval Parameters*

| Parameter | Description |
|---|---|
| start_time | A string specifying the start time to desired time in the standard format *dd-mmm-yyyy hh mm ss* 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 *dd-mmm-yyyy hh mm ss* 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<br><br>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 start index on subsequent calls This enables retrieval of all events within the time period even if the requested number of output values in a single call 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 EVENTT ME | The t me of the event | A str ng n the standard time format |
| EB EVENTTEXT | The text of the event | Two strings S nce each event can be a max mum of two nes two output va ues are used for each event f a g ven event has on y one ine of text then the second output value for the event s an empty str ng |
| EB EVENTTYPE | The type of the event | A str ng by default Append N to the attr bute name to have the num er c value returned<br>0 State change and a arm<br>1 Ana og a arm<br>2 Operator contro<br>3 Operator note<br>5 State change<br>6 – D gita alarm<br>7 – Operator conf gurat on<br>8 Alarm acknowledge<br>254 – T t e |
| EB TAGTYPE | The type of the tag assoc ated w th the event If an event is not associated with a tag then AAG NODATA value wi l be returned | Append N to the attribute name to have the numeric value returned |
| EB TAG | The name of the tag assoc ated with the event f an event is not assoc ated w th a tag then AAG NODATA va ue w l be returned | String 14 |
| EB ALMSTS | The a arm status of the tag assoc ated with the event at the time of the event at the t me of the event If an event is not assoc ated with a tag then AAG NODATA value w l be returned | Append N to the attr bute name to have the numer c va ue returned Refer to Table A 1 for va ues |
| EB QUAL | The qua ity of the tag assoc ated w th the event at the time of the event If an event is not associated w th a tag then AAG NODATA value w l be returned | Append N to the attr bute name to have the numeric value returned Refer to Table A-1 for va ues |
| EB LOOP | The hardware loop address of the tag associated w th the event at the time of the event f an event is not associated with a tag then AAG NODATA va ue w l be returned | nteger Refer to Tab e A 1 for values |
| EB NODE | The hardware node address of the tag assoc ated with the event at the time of the event f an event s not associated w th a tag then AAG NODATA va ue wi be returned | Integer Refer to Table A 1 for va ues |

*Table 4 8 Returnable Attributes (continued)*

| Attribute Names | Description | Return Values |
|---|---|---|
| EB MODULE | The hardware modu e address of the tag associated w th the event at the t me of the event If an event s not assoc ated with a tag then AAG NODATA va ue w be returned | Integer Refer to Tab e A 1 for va ues |
| EB BLOCK | The hardware b ock address of the tag associated with the event at the time of the event f an event is not associated w th a tag then AAG NODATA va ue will be returned | nteger Refer to Tab e A 1 for va ues |
| EB ALMGRP | The a arm 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 va ues |

The output values of this command depend on the input val ues 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 corre sponding 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 out put 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 for mat is *dd-mmm-yyyy hh mm ss* Referer ce time rep resents the o dest t me (to return) when d rect on *spec f ed is FORWARD t represents the most recent* t me when direction s BACKWARD |
| D rect on | A string specify ng the direction to move with respect to the start/reference time FORWARD means to move forward n t me beginn ng at the start/reference t me FORWARD cannot be used in conjunction w th *a start t me of CURRENT BACKWARD means to* move backward n time beg nn ng at the start/refer- ence 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 men tioned 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<br>10 20 30 | 28947 | Next start index |
| 10-MAY 1996<br>12 20 30 | | Pad to start of events |
| 0 | 10 20 3* A*00 *09 2H *ANK 1<br>LEVEL | L ne 1 of first matcn ng event |
| | TOO FULL | L ne 2 of first matching event |
| | 10 20 45 D100 OPEN VALVE 4 | L ne 1 of second match ng event |
| | | Line 2 of second matching event empty |
| | 10 21 05 A200 106 9 GAS FLOW | Line 1 of th rd matching event |
| | OK | L ne 2 of th rd match ng event |
| | | Empty, not enough output values rema n to f t all attr butes for another event |

*Figure 4 1 Get Events Example (Time Span Retrieval)*

**GetEvents**

| Input Values | Output Values | |
|---|---|---|
| 10 MAY 1996<br>10 20 30 | 10-MAY 1996 10 20 31 | Time of f rst matching event |
| FORWARD | 10 20 31 A100 109 2H TANK 1<br>_EVEL | Line 1 of first match ng event |
| EB EVENTIME | TOO FULL | Line 2 of f rst match ng event |
| EB EVENTEXT | 10 MAY-1996 10 20 45 | Time of second match ng event |
| | 10 20 45 D100 OPEN VALVE 4 | Line 1 of second match ng event |
| | | Line 2 of second matching event empty |
| | 10-MAY-1996 10 21 05 | Time of third match ng event |
| | 10 21 05 A200 106 9 GAS FLOW | L ne 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*

*Sheet1!A1* 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) v a wh ch the OIS 40 Ser es or Conductor VMS software s commun cating w th the X window server to display the conso e or workstat on w ndow |
| Tune Lock State | Boo ean | State of the configuration key ock assoc ated w th the conso e or workstation w ndow |
| Conf gurat on Lock State | Boo ean | State of the conf gurat on key lock assoc ated w th the console or workstation w ndow |

## *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 com plete 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 par ticular 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 acknowledg ment d alog 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 | nteger float, or str ng | Window number (can be a number coded as a string, but must be a number) |
| D sp ay Name | Str ng | Conso e or workstat on d sp ay name, up to e ght characters |
| T meout | Integer, f oat or string | Optiona input va ue T me-out per od for dia og box in seconds If t me-out s ess than or equa to zero then no time out s used and the command waits ndef nite y on an operator response No time out is the defau t |
| Prompt | Integer, f oat, or string | Optional nput value A prompt for the dia og box w th a maximum of 60 characters A default prompt s used f none s spec fied |

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 mes sage 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 mes sage 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 exam ple of a **DiagnosticLog** command input value. Numeric input values are converted to strings before being written to the con sole 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 |
|---|---|---|
| Va ue1 n | Integer, float, or string | Informat on to be written to the O S 40 Series and Conductor VMS d agnostic 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 eft corner coo rdinates |
| x2-coord y2 coord | Upper r ght corner coord nates |
| servername~ | Name of the user written @aG ance/ T server The server name can be either a ogica or actual server name If using a og ca name, it must appear in the system logical name ab e Max mum of 32 characters fol owed by a ti de (~) |
| commandname~ | Name of the server spec fic command to nvoke Max - mum ot 32 characters to lowed by a t de (~) |
| connect on | 0 disconnect<br>1 = keep connected |
| argument~ | A str ng to be passed with the command as one of the nput values Maximum of 80 characters fo owed 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 discon nect 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 communica tion on subsequent activations

### *Key Select Activation*

The @aGlance/IT user task activation via key select is config ured 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 | ASC key code for the second key |
| servername~ | Name of the user written @aGlance/IT server The server name can be either a og cal or act ja server name If using a logical name it must appear n the system logica name tab e Maximum of 32 characters followed by a t lde (~) |
| commandname~ | Name of the server spec fic command to invoke Max - mum of 32 characters fol owed by a ti de (~) |
| connect on | 0 disconnect<br>1 ~ keep conr ected |
| argument~ | A str ng to be passed w th the command as one of the nput va ues 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 discon nect 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 communica tion 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 specificat on f le |
| **GEN_TSF** | Generates a samp e command specification file This file conta ns comments (any ine beginning with an *) and command spec fication records |
| **HELP** | D sp ays a brief descript on of the command processor commands Th s command accepts no arguments (i e , f e spec) |
| **START** | Used to send a command to a server when the console software is started This com mand selects the command spec f cat on records n the command specificat on fi e whose *start* field conta ns *Y*[1] |
| **STOP** | Used to send a command to a server when the conso e software s shutdown This command se ects the command specif cation records in the command spec fication fi e whose *stop* field contains *Y*[1] |

**NOTE** 1 The CRT parameter of the command s set to zero to nd cate that th s s an activation by **AAG CP** not from a conso e d sp ay

*Table 4 19 AAG CP Command Specification File*

| Command | Contents | Description |
|---|---|---|
| start | Y or N | f Y this command is sent when the console software is started This fie d is case sensitive |
| stop | Y or N | If N th s command is sent when the conso e software s shutdown Th s f eld is case sensitive |
| server | Server name | Name of the user-written server task to which to send the command There is a 32 character l mit to this field This f eld is not case sensitive |
| command | Command name | Name of the command to send to the user-written server task The com mand must conform to the command layout guidelines contained in ***ACTIVATE REMOTE TASK*** There s a 32 character l mit to this field t may or may not be case sensitive depend ng on how the co mmand s defined n the server |
| parameter str ng | Up to an 80 character string | Opt ona string to pass along with the command Th s field can be left blank It can conta n 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 |
|---|---|
| Numer c ndex | Index of the f rst message to view n the log A zero mp es the most recent message |
| A C ients | Enable/d sable nformational message ogc ng for a c - ents based on the nput values that fo ow |
| [No] nformat ona | Enab e/disab e all nformat ona messages |
| [No]GetTags | Enable/d sable logging for GetTags re ated messages |
| [No]GetAttrs | Enab e/d sable ogg ng for GetAttrs re ated messages |
| [No]GetLIst | Enab e/disab e logg ng for GetL st re ated messages |
| [No]GetTab e | Enable/d sable logg ng for GetTable re ated messages |
| [No]PutL st | Enable/d sab e logging for PutL st re ated messages |
| [No]PutTab e | Enab e/d sable ogging for PutTable re ated messages |
| [No]GetH story | Enab e/disable ogg ng for GetHistory re ated messages |
| [No]GetH stEvent | Enable/disab e logg ng for GeH stEvent re ated messages |
| [No]GetH stStat | Enable/d sable logging for GetH stStat re ated messages |
| [No]Mon tor | Enable/d sable logging for mon tor ng related messages |

The following examples show how the server specif c **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

Examp e 1 Macro to enab e and disable diagnostics using the **AAG Command** input va ues opt on

**=AAG (' Command", *"yourserver"*, "Diagnose", *Sheet1!A1 A3,)***

This macro inputs cell values from Sheet1 where the spread sheet 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 | NoGetL st | | |
| 4 | | | |

Examp e 2 Macro to view latest d agnost c message log using the **AAG Command** out put values opt on

**=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 val ues option of the **AAG Command** places the diagnostic mes sages in the output areas set up on spreadsheet Sheet2

The Sheet2 spreadsheet containing output areas for this example:

| Sheet2 | |
|---|---|
| | A |
| 1 | Diagnost c message output area |
| 2 | Diagnostic message output area |
| 3 | Diagnostic message output area |
| 4 | D agnostic message output area |
| 5 | D agnostic message output area |
| 6 | D agnostic 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 va ue is interpreted as a date if the cel has a format of GENERAL

Example 3 Macro to scroll through d agnostic messages using **AAG Command** input and output va ues opt on

**=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 prohibited |
| 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 GetTab e, AAG PutList, AAG PutTable, AAG Monitor AAG GetH story AAG GetHistEvent, AAG GetHistStat | The tag name specif ed s not defined on the con sole on which the server s runn ng Note that tag names are case sens tive |
| Bad Type | AAG GetL st AAG GetTable AAG Mon tor | The attribute spec fied does not apply to the type of the given tag |
| Bad Value | AAG PutList AAG PutTab e | The given va ue cannot be converted to the expected data type of the attribute For example, a str ng containing non numeric characters is spec f ed for an nteger attribute |
| Bad Write | AAG PutL st, AAG PutTable | Unable to write tag configuration information to fi e |
| Disestabl shed | AAG GetList, AAG GetTable<br>AAG PutList AAG PutTab e, AAG Mon tor | The tag is off-scan |
| Mode nterlocked | AAG PutL st AAG PutTable | The STAT ON tag is mode locked and the mode and level cannot be changed |
| No Specs | AAG GetList, AAG GetTable, AAG Monitor | No specif cations have been rece ved for the tag and the spec fied attribute is an NFI90 spec f ca tion value |
| Not in Substitute | AAG PutList AAG PutTable | An attempt was made to wr te an attribute that s only substitutable but the tag s not n substitute The EB SUBST attribute must be written w th a va ue of one, directly followed by a substitutable attr bute in the same request in order to p ace the tag in substitute mode |
| Not Processed | AAG PutL st, 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 attr butes for the same tag is aborted when an error occurs to protect aga nst the tag being in an unexpected state when wr ting va ues |
| RED TAGGED | AAG PutList, AAG PutTable | The tag s red tagged and the *Suppress Control if Red Tagged* system configuration opt on s enabled so process control s nhib ted |
| Source Locked | AAG PutL st AAG PutTable | An attempt was made to change the nput source for a DANG or DADIG tag, but the correspond ng block is locked into the current input source |
| Subst tuted | AAG PutL st AAG PutTable | An attempt was made to write a process contro attribute when the tag s n substitute mode The tag must be placed on scan before a process con trol operation can be performed |
| TRACKING | AAG PutList AAG PutTable | An attempt was made to write to a value that s being tracked by NFI90, such as setting the set point on a STAT ON that is set point track ng |
| Trend not found | AAG GetH story AAG GetHistEvent, AAG GetHistStat | No trend was found match ng the g ven 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 g ven va ue to wr te is not with n the range of the expected va ues for the spec fied attr bute |
| Wrong Mode | AAG PutL st, AAG PutTab e | An attempt was made to perform a contro opera tion on a STAT ON tag when the STAT ON is not in the correct mode For example attempting to write the control output when the STATION s in AUTO mode |
| Wrong Stat on Type | AAG PutL st AAG PutTable | An attempt was made to put a STATION into RATIO or CASCADE mode but the station is not a RAT O or CASCADE stat on |
| Reso ut on nva id | AAG GetH story | The requested nterva t me (e g 15 seconds) is less than the collection resolution (e g , one m nute) of a trend in the request |
| Not In Yet | AAG GetHistory<br>AAG GetHistEvent<br>AAG GetH stStat | A request for data was made but the data has not been processed or received yet |
| Not Connected | AAG GetHistory<br>AAG GetHistEvent,<br>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 t also occurs if the tag index, trend mode, or tag subtype has been changed n 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 east one value used in a stat stica calcu at on had bad qua ity |
| Missing | AAG GetHistory,<br>AAG GetH stEvent,<br>AAG GetH stStat | Data is not ava lable on the console For standard trends, th s norma ly occurs for time per ods when the console or workstation is shutdown For enhanced trends, this w l depend on how long the module can store the data |
| Not Retr eved | AAG GetHistory<br>AAG GetHistEvent,<br>AAG GetHistStat | Data s not currently avai able on the conso e but s shou d be available due to a past archive to media The data must be retrieved using archive data retr eva procedures |
| Not Ava ab e | AAG GetHistory,<br>AAG GetHistEvent<br>AAG GetHistStat | Data s not currently avai able on the conso e and an attempt to retr eve the data from arch ve media was made but unsuccessful due to bad data on the archive med a |
| Data Error | AAG GetHistory,<br>AAG GetH stEvent,<br>AAG GetH stStat | Error encountered dur ng an attempt to read trend data from an arch ve media |
| nterna Error | AAG GetH story<br>AAG GetH stEvent<br>AAG GetH stStat | Some interna 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 partic ular tag/attribute pair is it is enabled resulting from unsuc cessful 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 |
|---|---|---|
| Al attributes nval d | AAG GetL st, AAG GetTable AAG PutL st, AAG PutTab e AAG Monitor, AAG GetH story AAG GetHistEvent, AAG GetHistStat | None of the attributes spec fied in the request are val d attribute names n the given context For example AAG PutL st reports this error f all attribute names are m sspe led or f none of the attributes are wr table, even if any of them are the name of a va d readable attribute |
| All tags inva id | AAG GetAttrs AAG GetL st AAG GetTable AAG PutList AAG PutTable, AAG Monitor AAG GetHistory, AAC GetH'stEvent, AAG GetHistStat | None of the tags specified in the request are defined on the console or workstation where the server is running |
| Bad d splay name spec fied | RequestDisplay | The display name input va ue is not a string |
| Bad statistical type | AAG GetHistStat | The stat stical type specif ed is not recognized The valid statistical types are<br>AAG H ST MAX<br>AAG HIST M N<br>AAG H ST 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 un t, such as a time-out on a busy loop |
| Client # (*n*) not found | ListClients | The client number specified exceeds the number of current y connected clients The List Clients commands only lists information on currently connected clients |
| Display name (*display*) invalid | RequestDisplay | The specified d splay was not found on the console or workstation where the server is running |
| Display of *display* on window *n* ignored | RequestDisplay | The operator rejected the d splay callup by choos ing the *Ignore* button in the acknowledgment dialog |
| Display of *display* on window *n* timed out | RequestDisplay | The operator did not acknowledge the d splay request within the time-out period spec fied in the request |

*Table 5 3 Error Messages (continued)*

| Return Error Text | Applicable Client Functions | Description |
|---|---|---|
| Error d splay ng d a- og box on w ndow *n* | RequestDisp ay | An error occurred on the conso e or workstat on when attempt ng to bring up the acknow edgment d a og for the d sp ay |
| Error encountered some data be ng returned | GetEvents | An error occurred wh e reading the event og snapshot file but data processed before the error w l be returned |
| Error send ng d sp ay | RequestD sp ay | An error occurred on the conso e or workstat on when attempt ng to callup the display |
| Event data not be ng saved to disk | GetEvents | The *System Event Log* conf guration is not setup to save events to disk Events must be saved to d sk for the GetEvents command to funct on |
| Forward option nva d with current or future time | GetEvents | The FORWARD opt on was spec f ed with a start t me for which no ater events are poss ble |
| Incorrect t me format | GetEvents | The format of the start or end time is incorrect The time must be a str ng in the form "DD-MMM-YYYY HH MM SS" or "CURRENT" |
| Inva d argument count | Diagnose DiagnosticLog GetW ndow, ListC ents, RequestD splay SetOpt ons | The number of input values does not conform to the requirements of the command |
| Inva d c ent # (*n*) specified | ListClients | The c ient number specif ed s not w th n the range of support c ients (1 5) |
| nval d monitor type requested | AAG Mon tor | The mon tor type spec fied s not recogn zed The va id monitor types are as fo lows<br>AAG POLLED<br>AAG EXCEPTION |
| nvalid time range | AAG GetHistEvent, AAG GetHistStat | The spec fied start t me is later than the specif ed end time |
| Inva d window # (*n*) specified | GetW ndow RequestDisp ay | The w ndow number specif ed is not with n the range supported by the console or workstation (1 8) |
| Ma lbox error | A l | A commun cat on error occurred on the conso e or workstation while attempt ng to perform the request |
| Max tags per mon tor request limit exceeded | AAG Mon tor | The request conta ns more than the maximum l mit of 100 tag attr bute pairs |
| Maximum mon tor request im t reached | AAG Mon tor | Each server supports up to 50 active monitor requests |
| Memory al ocation or system communica- t on fai ure | AAG GetList AAG GetTab e, AAG Monitor | There s insuffic ent memory on the conso e or workstation to perform the request or a commun - cation error occurred on the conso e or workstat on wh le attempting to perform the request |
| Monitor request CB connect fai ed | AAG Monitor | The max mum number of mon tored tags was exceeded Each server supports up to 250 mon - tored tags |

*Table 5 3 Error Messages (continued)*

| Return Error Text | Applicable Client Functions | Description |
|---|---|---|
| Monitor update fa ed abort ng mon tor request | AAG Monitor | An error occurred wh e attempt ng to update a monitor request View the server d agnost c og for addit ona information |
| No data match request cr teria | GetEvents | No events matching the nput t me range or opt on were found |
| No match ng attr butes found | AAG GetAttrs | No attr butes matching the g ven pattern from the given start index were found |
| No match ng tags found | AAG GetTags | No tags match ng the g ven pattern from the given start index were found |
| Output buffer too smal for request | GetEvents | The number of output values is insuff cient to return the selected (or default) attr butes for even a s ng e event |
| Requested monitor po nterva s inval c | AAG Monitor | The pol interval spec fied in the request is ess than the m n mum poll interva of two seconds |
| Server mon tor ng disab ed init a iza tion failed | AAG Monitor | The server encounter a memory a ocation error wh le trying to init a ze mon toring |
| Shared memory a lo- cat on error | A l | There s insuff cient memory on the conso e or workstation to perform the request |
| Start index no onger valid | GetEvents | The start index is no onger va d s nce a new event log snapshot was performed s nce the ast request |
| Start time newer than end time | GetEvents | The spec fied start t me is ater than the specified end t me |
| System busy, snap shot f e not ava lab e | GetEvents | The O S appl cat on is a ready perform ng an event log snapshot in response to a request from another client or cperator |
| System or file error | GetEvents | A memory a ocat on system communication, or fi e error occurred in the process ng of the request |
| System or f e error dur ng snapshot request | GetEvents | A memory a locat on, system communicat on or file error occurred n the processing of the snap- shot request |
| The C U is OFFLINE | AAG GetL st, AAG GetTable, AAG PutList, AAG PutTable, AAG Monitor, AAG GetHistory, AAG GetHistEvent AAG GetHistStat | The computer interface unit is off ine so no data s available |
| Time out (*n*) inva d | RequestD splay | The time-out nput value cou d not be converted to an nteger or t is ess than zero |
| too few or too many parameters | GetEvents | The number of input va ues does not conform to the requ rements of the command |
| Unable to get og messages from f le | Diagnose | An error occurred attempting to read the server d agnost c log fi e |
| Unable to open log file | Diagnose | An error occurred attempt ng to open the server diagnostic log f le |

*Table 5 3 Error Messages (continued)*

| Return Error Text | Applicable Client Functions | Description |
|---|---|---|
| Unknown command %s | AAG Command | The command str ng spec f ed s not one of the fo ow ng command str ngs recogn zed by the Elsag Ba ey @aG ance/IT server<br><br>D agnost cLog<br>GetEvents<br>GetW ndow<br>L stC ents<br>RequestD splay<br>SetOpt ons |
| Unrecognized input parameter *n* | GetEvents | The *nth* input value does not conform w th the requirements of the command |
| Window # *n* nactive | GetWindow, RequestDisp ay | The console or workstat on w ndow is not act ve |

# 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 attr butes are spec fic to Harmony /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 termi nator. 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 com ment 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 | Descript on | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C DESC | Tag Descr pt on | R | String 32 | Tag Description |
| EB ALMSTS | Current alarm status | R/W/S[2] | nteger<br>String 2 | 0 blank no a arm<br>1 A D gita N90STA or TEXSTR a arm<br>2 * Bad qua ty<br>3 N Return to norma<br>4 H gh a arm<br>5 L Low alarm<br>6 HD H gh deviation a arm<br>7 LD Low dev at on a arm<br>8 2H Two h gh a arm<br>9 2L Two low a arm<br>10 3H Three h gh a arm<br>11 3L Three low a arm<br>12 HR H gh rat a arm<br>13 LR Low rate a arm<br>14 HF Hardware a arm<br>15 Not used<br>16 A arm suppressed |
| EB BADQUAL | Good/Bad quality | R/W | Boo ean<br>Str ng 4 | 0 – GOOD<br>1 BAD |
| EB BLOCK | Tag hardware block address | R | nteger | 0 9998 |
| EB CURCMT | A arm comment for current a arm state | R | String 64 | User conf gured str ng |
| EB CUPR O | A arm priority for current a arm state | R | nteger<br>Str ng 8 | 1 8 1 has the h ghest priority<br>1 DIRE<br>2 CRUCIAL<br>3 CRIT CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 – NFORM |
| EB CUSTTAGID | Customer tag D | R | Str ng 32 | Str ng from tag datab ase |
| EB C NDEX | C ndex number | R | Integer | 1 30000 |
| EB LOOP | Tag hardware oop address | R | nteger | 0 250 |
| EB MAN NHB | A arm manua y inh bited flag | R/W | Boolean | 0 – no manual nh b t<br>1 manua nh b t |
| EB MODULE | Tag hardware modu e address | R | nteger | 0 31 |

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

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB NODE | Tag hardware node address | R | nteger | 0 250 |
| EB PRDSP | Pr mary d splay name | R | Str ng 8 | B ank padded d sp ay name |
| EB PRDSPCRT | CRT on which to callup pr mary display | R | Integer | 0 8 0 is the current d sp ay |
| EB QUAL | Qual ty | R/W | nteger<br>Str ng 11 | 0 b ank Good<br>1 * Bad<br>2 x Off scan<br>3 s Subst tuted<br>4 ? Suspect<br>5 I nh b t |
| EB SCANSTS | Scan status | R/W | Boo ean | 0 off scan<br>1 on scan |
| EB SUBST | Value s substituted va ue | R/W | Boo ean | 0 – not subst tuted<br>1 substituted |
| EB TAG | Tagname | R | String 14 | User conf gured str ng |
| EB TAGTYPE | Tag type | R | Integer<br>String 8 | 0 – UNDEF<br>1 ANALOG<br>2 D GITAL<br>3 – N90STA<br>4 – RCM<br>5 – STAT ON<br>6 – RMSC<br>7 DD<br>8 MSDD<br>9 – RMCB<br>10 DAANALG<br>11 DAD GTL<br>12 – TEXT<br>13 DANG<br>14 DAD G<br>15 TEXTSTR<br>16 = INTANG<br>17 NTD G<br>18 DEVSTAT |
| EB XRTIMESTAMP | Exception report t me stamp | R | Strng 20 or<br>String 23 | DD MMM YY HH MM SS<br>or<br>DD MMM-YY HH MM SS XX<br>f the modu e time stamp feature is enabled |

**NOTES**
1 nd cates the operations that can be performed on the tag read (R) write (W) and substitute (S)
2 Can write to NTANG (non H L) and NTD G (none A) tag types Can subst tute TEXTSTR (none A) tag types
3 Can write to analog report (GOOD BAD) digita report (GOOD BAD) nterna digita (GOOD BAD) and nterna analog (GOOD BAD) tag types
4 Can write to ana og report d g ta report nterna d gita nterna analog and text strng tag types

*Table A 2 ANALOG Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C APV | Analog va ue | R/S | F oat | Expressed n eng neering un ts |
| C PVH AP | High a arm mit value | R | F oat | Expressed n eng neering un ts |
| C PVLOAP | Low alarm mit value | R | F oat | Expressed in eng neer ng units |
| C RANGLO | Process variable zero scale | R | F oat | Expressed n engineer ng units |

*Table A 2 ANALOG Tag Attributes (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C UN TS | Eng neer ng un ts descr ptor | R | Integer<br>Str ng 8 | A l blanks f not ava able<br>0 - 255 ndex into tab e of eng *neer ng descr ptors The str ng* returned s the descr ptor n the tab e corresponding to the ndex |
| EB CALIBSTS | Ca bration status | R | Boo ean<br>String 14 | 0 NORMAL<br>1 OUT OF RANGE |
| EB CONF GERR | Configuration error ax sts f ag | R | Boo ean<br>Str ng 14 | 0 CONF G OK<br>1 CONFIG ERROR |
| EB D SABLED | Po nt d sabled | R | Boo ean<br>Stnng 14 | 0 NORMAL<br>1 SERVICED |
| EB OTYPE | D stingu shes between standard and enhanced /O types | R | nteger<br>Str ng 7 | 0 STD<br>1 ENH/ N<br>2 ENH/OUT |
| EB OVR | Override enable flag | R/W | Boolean<br>Stnng 14 | 0 NORM<br>1 OVR |
| EB OVRPERM | Override perm ss ve flag | R | Boolean<br>Str ng 14 | 0 OVR PERMITTED<br>1 OVR LOCKED OUT |
| EB OVRVAL | Overr de value | R/W | F oat | float va ue |
| EB PROPQUAL | Propagated qua ty | R | Boo ean<br>Str ng 14 | 0 GOOD QUAL PROP[2]<br>1 BAD QUAL PROP[3] |
| EB RANGE | Range Block /O channe cond tion | R | nteger<br>Str ng 14 | 0 OK<br>1 OPEN<br>2 SHORT<br>3 OVERDR VE/LOW<br>4 OVERDR VE/H C H |
| EB READBACK | Readback Status | R | Boo ean<br>Str ng 14 | 0 READBK STAT OK<br>1 READBK STA BAD |
| *EB REFSTS* | *Reference Status* | *R* | *Boolean*<br>Stnng 14 | *0 REF STATUS OK*<br>1 REF STATUS BAD |
| EB S MMODE | S mulation mode flag | R | Boolean<br>Str ng 14 | 0 NORMAL MODE<br>1 S M MODE |
| EB SPAN | Sca e span | R | F oat | Expressed in eng neer ng un ts |
| EB STSERR NH | Status errors inh b t ed f ag | R | Boolean<br>String 14 | 0 NORMAL<br>1 STAT ERR NH B |

**NOTES**
1 nd cates the operations that can be performed on the tag read (R) wnte (W) and subst tute (S)
2 GOOD QUAL PROP is only meaningful for the enhanced I/O types This va ue means the propagated qua ty received s good
3 BAD QUAL PROP s on y meaningfu for the enhanced I/O types Th s va ue means the propagated qua ity received s bad

*Table A 3 DAANALG Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C ALMDB | A arm deadband | R | Float | Expressed n eng neering un ts |
| C APV | Analog va ue | R/S | Float | Expressed n eng neering un ts |
| C PVHIAP | High alarm limit value | R | Float | Expressed in engineering units |
| C PVLOAP | Low a arm m t va ue | R | Float | Expressed n eng neering un ts |
| C RANGLO | Process var able zero sca e | R | F oat | Expressed n eng neering units |

*Table A 3 DAANALG Tag Attributes (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C UN TS | Eng neer ng un ts descr ptor | R | Integer Str ng 8 | Al b anks f not ava able 0 255 ndex nto tab e of eng neer ng descriptors The str ng returned s the descr ptor n the tab e correspond ng to the ndex |
| EB H 2DELTA | H gh h gh a arm m t va ue | R | F oat | Expressed n eng neer ng un ts |
| EB H 3DELTA | H gh h gh h gh a arm m t value | R | F oat | Expressed n eng neer ng units |
| EB H VARALMTAG | Vanab e h gh a arm m t tag | R | Str ng 14 | B ank padded tag name |
| EB LO2DELTA | Low ow a arm l m t va ue | R | F oat | Expressed n eng neer ng un ts |
| EB LO3DELTA | Low ow low alarm m t va ue | R | F oat | Expressed n eng neer ng un ts |
| EB LOVARALMTAG | Variable low a arm limit tag | R | String 14 | B ank padded tag name |
| EB SPAN | Scale span | R | F oat | Expressed n eng neer ng un ts |

**NOTE**
1 nd cates the operations that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 4 DADIG Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C DPV | D gita state | R/S | Boolean | 0 zero state<br>1 one state<br>The value returned or subst tuted s determined by EB SRC |
| C SPV | Log c state descr ptor of C DPV | R | String 16 | IF C DPV ts<br>0 then string contents of EB LS0<br>1 then str ng contents of EB LS1 |
| EB ALMSUP | Enab e/disable a arm suppress on | W | Boo ean | 0 unsuppress a arms<br>1 – suppress a arms |
| EB CALCST | Calculated (a ternate) state[2] | R | Boolean | 0 zero state<br>1 one state |
| EB FORCEXR | Force an exception report | W | Boo ean | 0 not used<br>1 – force an except on report |
| EB LATCHED | Value/status atched flag | R/W | Boolean Str ng 14 | 0 – UNLATCHED<br>1 = LATCHED |
| EB LS0 | Zero state descriptor | R | Str ng 16 | Logic state descriptor for zero state |
| EB LS1 | One state descriptor | R | String 16 | Logic state descr ptor for one state |
| EB NOREPORT | Modu e level no report flag | R/W | Boolean Str ng 14 | 0 ON REPORT<br>1 NO REPORT |
| EB PERINSEL | Permit nput source select on f ag | R | Boo ean, String 14 | 0 – LOCKED<br>1 UNLOCKED |
| EB QUALOVR | Qua ty overridden flag | R | Boo ean String 14 | 0 – NORMAL<br>1 OVERRIDDEN |
| EB REALM | Re a arm toggle | R | Boo ean, String 14 | 0 ZERO<br>1 ONE |
| EB REALST | Rea (primary) state[3] | W | Boo ean | 0 – zero state<br>1 one state |
| EB REDTAG | Red tagged | R | Boolean Str ng 14 | 0 NORMAL<br>1 RED TAGGED |
| EB SRC | Source of the reported value | R/W | nteger Str ng 14 | 0 USER INSERTED<br>1 ALTERNATE<br>2 PR MARY |

*Table A 4 DADIG Tag Attributes (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB USERST | User nserted state[2] | R/W | Boolean | 0 zero state<br>1 one state |
| EB OUTRANGE | Determ nes f the value is uncon strained and out of range | R | Boo ean<br>Str ng 12 | 0 N RANGE<br>1 OUT OF RANGE |
| EB USERTYPE | User type code | R | nteger | 0 255 |

**NOTES**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)
2 Th s attr bute causes the va ue to be read from the block Use it to obta n the calcu ated va ue when the DAD G s not n a ternate mode
3 This attribute causes the value to be read from the block Use it to obtain the real value when the DADIC s not in primary mode

*Table A 5 DAANG (DANG Tag Type) Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C APV | D g ta state | R/W/S | Float | Expressed in eng neering un ts |
| C UNITS | Eng neer ng units descriptor | R | Integer<br>String 8 | All b anks f not available<br>0 255 ndex into table of eng neering descr ptors The string returned is the descriptor in the tab e correspond ng to the ndex |
| EB ALMSUP | Enable/disable a arm suppression | W | Boo ean | 0 d sab ed<br>1 enab ed |
| EB CALCVAL | Calcu ated va ue | R | F oat | Expressed n eng neering units |
| EB CNTRREF | Center disp ay reference va ue | R | F oat | Expressed n eng neering un ts |
| EB CONSTR | Va ue constra ned f ag | R | Boolean<br>Str ng 14 | 0 UNCONSTRA NI D<br>1 CONSTRA NED |
| EB CONSTRPER | Constraint m ts enab ed flag | R | Boo ean<br>String 14 | 0 D SABLED<br>1 ENABLED |
| EB FORCEXR | Force an exception report | W | Boolean | 1 force except on report |
| EB H CONSTRLMT | H gh constra nt limit va ue | R | F oat | Expressed n eng neering units |
| EB H REF | High d sp ay reference va ue | R | Float | Expressed n eng neer ng un ts |
| EB LOCONSTRLMT | Low constra nt m t va ue | R | F oat | Expressed in engineering un ts |
| EB LOREF | Low d sp ay reference value | R | F oat | Expressed in engineering un ts |
| EB MULTLVL | Mult -leve alarm ng enabled flag | R | Boo ean<br>String 14 | 0 D SABLED<br>1 ENABLED |
| EB NOREPORT | Modu e level no report f ag | R/W | Boo ean<br>String 14 | 0 ON REPORT<br>1 NO REPORT |
| EB NXTH LMT | Next h gher m t | R | F oat | Expressed n eng neering units |
| EB NXTLOLMT | Next ower m t | R | Float | Expressed n eng neering un ts |
| EB PERINSEL | Permit nput source se ect on flag | R | Boo ean<br>String 14 | 0 LOCKED<br>1 ~ UNLOCKED |
| EB QUALOVR | Qual ty overr dden flag | R | Boolean,<br>String 14 | 0 NORMAL<br>1 OVERRIDDEN |
| EB REALM | Re-a arm togg e | R | Boolean<br>String 14 | 0 ZERO<br>1 ~ ONE |
| EB REALVAL | Rea state | W | Boo ean | Expressed n eng neering un ts |
| EB REDTAG | Red tagged | R | Boo ean<br>String 14 | 0 ~ NORMAL<br>1 RED TAGGED |
| EB SRC | Source of the reported va ue | R/W | nteger<br>Str ng 14 | 0 USER INSERTED<br>1 ALTERNATE<br>2 PR MARY |
| EB USERVAL | User nserted state | R/W | Float | Expressed n eng neer ng un t |

*Table A 5 DAANG (DANG Tag Type) Tag Attributes (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB OUTRANGE | H gh or ow constra nt m ts exceeded status | R | Boo ean Str ng 14 | 1 OUT OF RANGE<br>0 N RANGE |
| EB HARDFAULT | Hardware fau t status | R | Boo ean Str ng 14 | 1 HARDWARE FA L<br>0 OK |
| EB VARALM | Var able a arms enab ed f ag | R | Boo ean Str ng 14 | 0 D SABLED<br>1 ENABLED |

**NOTE**
1 nd cates the operat or s that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 6 DADIGTL/INTDIG Tag Attributes*

| Attribute[1] | Description | R/W/S[2] | Data Type | Value |
|---|---|---|---|---|
| C DPV | D gita state | R/W/S | Boo ean | 0 zero state<br>1 one state |
| C SPV | Symbolic state | R | Str ng 16 | F C DPV s<br>0 then string contents of EB LS0<br>1 then string contents of EB LS1 |
| EB ALMDEF | Alarm state defin tion | R | nteger Str ng 8 | 0 ZERO<br>1 ONE<br>2 NE THER |
| EB LS0 | Zero state descriptor | R | Str ng 16 | Log c state descr ptor for zero state |
| EB LS1 | One state descr ptor | R | Str ng 16 | Log c state descr ptor for one state |

**NOTES**
1 Only C DPV and C SPV apply to the O S 40 Series conso e or Conductor VMS workstat on for the NTD G attribute
2 nd cates the operatio ns 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 | Digita state | R/W/S[2] | Boo ean | 0 reset contro output<br>1 set contro output |
| C SPV | Symbo c state | R/W[3] | Str ng 16 | F C DPV is<br>0 then string contents of EB LS0<br>1 then string contents of EB LS1 |
| EB DDMODE | Device driver mode | R/W[4] | nteger Str ng 6 | 0 AUTO<br>1 – REMOTE<br>2 MANUAL |
| EB FB1 | Feedback 1 state | R | Boolean String 16 | 0 – EB FB1LS0<br>1 EB FB1LS0 |
| EB FB1LS0 | Feedback 1 zero state descriptor | R | Str ng 16 | Log c state descr ptor for feedback 1 zero state |
| EB FB1LS1 | Feedback 1 one state descr ptor | R | Str ng 16 | Logic state descr ptor for feedback 1 one state |
| EB FB2 | Feedback 2 state | R | Boolean Str ng 16 | 0 EB FB2LS0<br>1 – EB FB2LS0 |
| EB FB2LS0 | Feedback 2 zero state descr ptor | R | Str ng 16 | Logic state descriptor for feedback 2 zero state |
| EB FB2LS1 | Feedback 2 one state descr ptor | R | Str ng 16 | Logic state descr ptor for feedback 2 one state |
| EB FBSTS | Feedback status | R | Boolean, Str ng 14 | 0 NORMAL<br>1 BAD |
| EB LS0 | Zero state descriptor | R | String 16 | Log c state descr ptor 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 descr ptor | R | String 16 | Log c state descr ptor for C DPV one state |
| EB REDTAG | Red tagged | R | Boo ean<br>Strng 14 | 0 NORMAL<br>1 RED TAGGED |
| EB STSOVR | Status overr de flag | R | Boolean,<br>String 14 | 0 NORMAL<br>1 OVERR DDEN |

**NOTES**
1 nd cates the operat ons that can be performed cn the tag read (R) wr te (W) and substitute (S)
2 Can only wr te in manual mode
3 O S 40 Ser es conso e and Cond_ctor VMS workstat on cannot write
4 Can only wr te auto and manua

*Table A 8 Digital Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C DPV | Dig ta state | R/S | Boo ean | 0 zero state<br>1 one state |
| C SPV | Symbolic state | R | Str ng 16 | IF C DPV is<br>0 then str ng contents of EB LS0<br>1 then string contents of EB LS1 |
| EB ALMDEF | A arm state defin tion | R | nteger,<br>String 8 | 0 ZERO<br>1 – ONE<br>2 NE "HEH |
| EB CONF GERR | Configurat on error ex sts flag | R | Boo ean<br>Str ng 14 | 0 CONFIG OK<br>1 CONFIG ERROR |
| EB OTYPE | D stingu shes between standard and enhanced I/O types | R | Boolean<br>Str ng 14 | 0 STD<br>1 ENH/ N<br>2 ENH/OUT |
| EB LS0 | Zero state descriptor | R | String 16 | Log c state descriptor for zero state |
| EB LS1 | One state descriptor | R | String 16 | Log c state descr ptor for one state |
| EB OVR | Override enab e f ag | R/W | Boo ean<br>String 6 | 0 NORM<br>1 OVR |
| EB OVRPERM | Override perm ssive flag | R | Boo ean<br>String 14 | 0 OVR PERM TTED<br>1 OVR LOCKED OUT |
| EB OVRST | Override state | R/W | Boo ean<br>String 6 | 0 <zero og c state descr ptor (EB LS0)><br>1 <one og c state descr ptor (EB LS1)> |
| EB PROPQUAL | Propagated qua ity | R | Boo ean<br>Str ng 14 | 0 GOOD QUAL PROP[2]<br>1 BAD QUAL PROP[3] |
| EB RANGE | Range B ock /O channe cond tion | R | nteger<br>String 14 | 0 OK<br>1 OPEN<br>2 – SHORT<br>3 – OVERDR VE |
| EB READBACK | Readback Status | R | Boolean<br>String 14 | 0 READBK STAT OK<br>1 READBK STA BAD |
| EB S MMODE | S mu ation mode flag | R | Boo ean<br>Str ng 14 | 0 NORMAL MODE<br>1 S M MODE |
| EB STSERR NH | Status errors nhibited flag | R | Boo ean<br>String 14 | 0 NORMAL<br>1 STAT ERR INH B |

**NOTES**
1 nd cates the operat ons that can be performed on the tag read (R) write (W) and subst tute (S)
2 GOOD QUAL PROP s only mean ngfu for the enhanced /O types Th s va ue means the propagated qua ty received s good
3 BAD QUAL PROP s only mean ngfu for the enh anced I/O types Th s va ue means the propagated qua ity received s bad

*Table A 9 INTANG Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C APV | Ana og va ue | R/W/S | F oat | Expressed n eng neer ng un ts |
| C PVH AP | H gh a arm m t va ue | R | F oat | Expressed n eng neering un ts |
| C PVLOAP | Low a arm m t va ue | R | F oat | Expressed n eng neer ng un ts |
| C RANGLO | Process var able zero sca e | R | F oat | Expressed n eng neer ng un ts |
| C UN TS | Eng neer ng un ts descnptor | R | nteger<br>Str ng 8 | A blanks f not ava ab e<br>0 255 ndex nto tab e of eng neering descr ptors The str ng returned is the descr ptor n the tab e correspond ng to the index |
| EB CAL BSTS | Ca bration status | R | Boo ean<br>Str ng 14 | 0 NORMAL<br>1 OUT OF RANGE |
| EB D SABLED | Po nt disabled | R | Boo ean<br>String 14 | 0 NORMAL<br>1 SERVICED |
| EB SPAN | Scale span | R | F oat | Expressed n eng neering un ts |

**NOTES**
1 nd cates the operat ons 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 | D gita state of contro output 1 | R | Boo ean | o Contro output zero<br>1 Contro output one |
| C PV | Last good state | R/S[2] | nteger | 0 3 |
| C ISP | Requested state | R/W[3] | nteger | 0 3<br>Can only wr te 1 3 |
| C SPV | Symbolic last good state | R | String 16 | f C PV s<br>0 then str ng contents of EB LS0<br>1 then str ng contents of EB LS1<br>2 then str ng contents of EB LS2<br>3 then str ng - contents of EB LS3 |
| EB AUTOMAN | Auto manua mode | R/W | Boo ean<br>String 6 | 0 MANUAL<br>1 AUTO |
| EB CNTRLOVR | Contro override flag | R | nteger<br>Str ng 14 | 0 - NORMAL<br>1 OvERR DDEN |
| EB FB1 | Feedback 1 state | R | Boo ean<br>String 16 | 0 EB FB1LS0<br>1 EB FB1LS1 |
| EB FB1LS0 | Feedback 1 zero state descnptor | R | String 16 | Logic state descriptor for zero state |
| EB FB1LS1 | Feedback 1 one state descr ptor | R | Str ng 16 | Log c state descr ptor for one state |
| EB FB2 | Feedback 2 state | R | Boolean<br>String 16 | 0 - EB FB2LS0<br>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 descnptor | R | Str ng 16 | Logic state descr ptor for one state |
| EB FB3 | Feedback 3 state | R | Boolean,<br>String 16 | 0 EB FB3LS0<br>1 EB FB3LS1 |
| EB FB3_S0 | Feedback 3 zero state descriptor | R | String 16 | ng c state descriptor for zero state |
| EB FB3LS1 | Feedback 4 one state descr ptor | R | Str ng 16 | Logic state descr ptor for one state |
| EB FB4 | Feedback 4 state | R | Boo ean<br>String 16 | 0 EB FB4LS0<br>1 - EB FB4LS1 |
| EB FB4LS0 | Feedback 4 zero state descriptor | R | Str ng 16 | Logic state descriptor for zero state |

*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 descr ptor | R | Str ng 16 | Log c state descr ptor for one state |
| EB FBSTS | Feedback status | R | Boolean<br>Strng 14 | 0 NORMAL<br>1 BAD |
| EB LS0 | Zero state descriptc r | R | Str ng 16 | Log c state descr ptcr for zero state |
| EB LS1 | One state descriptor | R | Str ng 16 | Log c state descr ptor for one state |
| EB LS2 | Two state descr ptor | R | Str ng 16 | Log c state descr ptcr for zero state |
| EB LS3 | Three state descnptor | R | Strng 16 | Logic state descnptor for one state |
| EB REDTAG | Red tagged | R | Boo ean<br>Str ng 14 | 0 NORMAI<br>1 RED TAGGED |
| EB STSOVR | Status overr de flag | R | Boo ean,<br>Str ng 14 | 0 NORMAL<br>1 OVERRIDDEN |

**NOTES**
1 Indicates the operat ons that can be performed on the tag read (R) write (W) and substitute (S)
2 Can on y subst tute 1 through 3
3 Can only wr te 1 through 3

*Table A 11 N90STA Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C BACCFG | Redundancy config ured | R | Boolean | 0 no redundancy<br>1 redundancy |
| EB BACSTS | Backup status flag | R | Boo ean<br>Str ng 14 | 0 GOOD<br>1 BAD |
| EB BACTAG | Tag name for redundant module | R | Str ng 14 | B ank padded tag name |
| EB LOC OSTS | Loca /O error f ag | R | Boolean<br>String 14 | 0 nor error<br>1 error |
| EB MODCFGERR | Modu e conf gurat on error f ag | R | Boo ean | 0 no error<br>1 error |
| EB MODERR | Modu e errors exist flag | R | Boo ean | 0 no error<br>1 error |
| EB MODMODE | Module mode | R | nteger<br>String 6 | 0 CONF GURE<br>1 FA LED<br>2 ERROR<br>3 EXECUTE<br>4 STANDBY[2] |
| EB MODMODEERR | Module mode error flag | R | Boolean | 0 error<br>1 no error |
| EB MODTYP | Module type | R | Integer<br>Str ng 8 | 1 P M<br>2 CTM |
| EB NCFGERR | Node configurat on error f ag | R | Boolean | 0 error<br>1 no error |
| EB NCOMMERR | Node communicat cn error f ag | R | Boo ean | 0 error<br>1 no error |
| EB NMODERR | Module errors n a r ode ex st f ag | R | Boo ean | 0 error<br>1 no error |
| EB NOFFL NE | Off l ne flag | R | Boo ean | 0 on ne<br>1 off ne |
| EB NSYSERR | Node system status error f ag | R | Boo ean | 0 error<br>1 no error |
| EB REMIOSTS | Remote I/O status | R | Boolean<br>Strng 14 | 0 NO ERROR<br>1 ERROR |

*Table A 11 N90STA Tag Attributes (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB SECONDARY | Pr mary/secondary flag | R | Boo ean | 0 pr mary mode<br>1 secondary mode |

**NOTES**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tut ə (S)
2 Standby status s equ va ent to a modu e n execute mode and secondary n a redundant pa r of modu es

*Table A 12 Remote Control Memory (RCM) Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C DPV | D g ta state | R/W/S | Boo ean | 0 zero state<br>1 one state |
| C SPV | Symbo c state | R | Str ng 16 | F C DPV s<br>0 then string contents of EB LS0<br>1 then str ng contents of EB LS1 |
| EB FEEDBACK | Feedback state ind cat ng the actual state of the contro led device | R | Boo ean<br>String 16 | 0 EB LS0<br>1 EB LS1 |
| EB LS0 | Zero state descnptor | R | String 16 | Log c state descr ptor for zero state |
| EB LS1 | One state descr ptor | R | String 16 | Logic state descr ptor for one state |
| EB OUTOVR | Output overr de flag | R | Boo ean<br>String 14 | 0 NOT SET<br>1 SET |
| EB RED"AG | Red tagged | R | Boo ea~<br>Str ng 14 | 0 NORMAL<br>1 RED TAGGED |
| EB RESCMD | Reset command | R | Boo ean,<br>String14 | 0 NOT SET<br>1 SET |
| EB RESIN | Reset input va ue | R | Boo ean<br>Str ng14 | 0 NOT RESET<br>1 RESET |
| EB SETCMD | Set command | R | Boo ean,<br>String14 | 0 NOT SET<br>1 SET |
| EB SET N | Set nput va ue | R | Boolean<br>Str ng14 | 0 NOT SET<br>1 SET |
| EB SETPER | Set permissive flag | R | Boolean<br>Str ng14 | 0 NOT SET<br>1 SET |

**NOTE**
1 nd cates the operations that can be performed on the tag read (R) wr te (W) and substitute (S)

*Table A 13 RMCB Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C DPV | Digita state of contro output 1 | R/W/S | Boolean | 0 – zero state<br>1 one state |
| C SPV | Symbolic last good state | R/W[2] | Str ng 16 | f C DPV s<br>0 then str ng – contents of EB LS0<br>1 then str ng – contents of EB LS1 |
| EB BADSTART | Bad start flag | R | Boo ean<br>Str ng 14 | 0 GOOD<br>1 – BAD |
| EB ERRCODE | Error code | R | nteger<br>Str ng 20 | 0 NO ERROR<br>1 STOPPED<br>2 NTERLOCK 1 LOW<br>3 INTERLOCK 2 LOW<br>4 INTERLOCK 3 LOW<br>5 INTERLOCK 4 LOW<br>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 | nteger<br>Str ng 20 | 7 FEEDBACK 2 LOW<br>8 FEEDBACK 1 H GH<br>9 FEEDBACK 2 H GH |
| EB FAULT | Fau t f ag | R | Boo ean<br>Str ng 14 | 0 NO FAULT<br>1 FAULT |
| EB FB1 | Feedback 1 state | R | Boo ean<br>Str ng 16 | 0 EB FB1LS0<br>1 EB FB1LS1 |
| EB FB1LS0 | Feedback 1 zero state descriptor | R | Str ng 16 | Log c state descr ptor for zero state |
| EB FB1LS1 | Feedback 1 one state descr ptor | R | String 16 | Log c state descr ptor for one state |
| EB FB2 | Feedback 2 state | R | Boo ean<br>Str ng 16 | 0 EB FB2LS0<br>1 EB FB2LS1 |
| EB FB2LS0 | Feedback 2 zero state descriptor | R | Str ng 16 | Log c state descr ptcr for zero state |
| EB FB2LS1 | Feedback 2 one state descr ptor | R | Str ng 16 | Log c state descriptor for one state |
| EB LS0 | Zero state descr ptor | R | String 16 | Logic state descr ptcr for zero state |
| EB LS1 | One state descr ptor | R | String 16 | Log c state descr ptor for one state |
| EB PM1LS0 | Perm ssive 1 zero state descr ptor | R | Str ng 16 | Log c state descr ptcr for zero state |
| EB PM1LS1 | Permissive 1 one state descr ptor | R | String 16 | Log c state descr ptor for one state |
| EB PM2LS0 | Perm ssive 2 zero state descr ptor | R | Str ng 16 | Logic state descr ptcr for zero state |
| EB PM2LS1 | Permiss ve 2 one state descriptor | R | Str ng 16 | Logic state descr ptor for one state |
| EB RED"AG | Red tagged | R | Boo ean,<br>Str ng 14 | 0 NORMAL<br>1 RED TAGGED |
| EB STARTPM1 | Start perm ss ve 1 state | R | Boolean<br>String 16 | 0 EB PM1LS0<br>1 EB PM1LS1 |
| EB STARTPM2 | Start perm ssive 2 state | R | Boolean<br>Str ng 16 | 0 EB PM2LS0<br>1 EB PM2LS1 |

**NOTES**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and substitute (S)
2 On y read (no wr te) for O S 40 Series conso e and Conductor VMS workstat on

*Table A 14 RMSC Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C APV | Analog va ue/set point | R/W/S | Float | Expressed n eng neer ng units |
| EB OPH LM | H gh im t va ue | R | Float | Expressed n eng neer ng un ts |
| EB OPLOLM | Low l m t value | R | F oat | Expressed n eng neer ng un ts |
| C RANGLO | Process var able zero sca e | R | Float | Expressed n eng neer ng units |
| C UN TS | Engineering un ts descr ptor | R | nteger<br>Str ng 8 | A blanks f not ava ab e<br>0 255 ndex into table of eng neer ng descriptors The str ng returned s the descr ptor n the tab e corresponding to the index |
| EB TRACKING | Output track ng f ag | R | Boolean,<br>Str ng 14 | 0 NO TRACK<br>1 TRACKING |
| EB SPAN | Sca e span | R | F oat | Expressed in engineering un ts |

**NOTE**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 15 Station Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| C AOUTPUT<br>EB VAL3 | Control output (CO) | R/W/S | Float | Expressed in engineering units |
| C APV<br>EB VAL1 | Analog process variable (PV) | R/S | Float | Expressed in engineering units |
| C ASP<br>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 (R) | R/W/S | Float | 0.0 if station is not in ratio mode |
| C UNITS | Engineering units descriptor | R | Integer<br>String 8 | All blanks if not available<br>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<br>String 6 | 0 MANUAL<br>1 AUTO |
| EB BYPASS | Bypass flag | R | Boolean<br>String 14 | 0 NO BYPASS<br>1 BYPASS |
| EB CMPTRSTS | Computer OK status flag | R/W | Boolean<br>String 14 | 0 FAILED<br>1 GOOD |
| EB COMMANDOUT[2] | Command output value/mode | W | Integer<br>String 32 | 0 LOCAL MANUAL<br>1 LOCAL AUTO<br>2 LOCAL CASCADE<br>3 COMPUTER MANUAL<br>4 COMPUTER AUTO<br>5 COMPUTER CASCADE<br>6 LOCAL LEVEL<br>7 COMPUTER LEVEL<br>8 COMPUTER BACKUP<br>9 COMPUTER OK<br>10 PREVIOUS STATE |
| EB DEVLMT | Deviation alarm limit value | R | Float | Expressed in engineering units |
| EB MODELOCK | Manual mode interlock flag | R | Boolean<br>String 14 | 0 NO MODE LOCK<br>1 MODE LOCK |
| EB OUTSTS | Station analog output status | R | Boolean<br>String 14 | 0 GOOD<br>1 BADE |
| EB OUTTRACKNG | Output tracking flag | R | Boolean<br>String 14 | 0 NO TRACK<br>1 TRACK NG |
| 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<br>String 14 | 0 GOOD<br>1 BAD |
| EB REDTAG | Red tagged | R | Boolean<br>String 14 | 0 NORMAL<br>1 RED TAGGED |
| EB SPTRACKNG | Setpoint tracking enabled flag | R | Boolean<br>String 14 | 0 NO TRACK<br>1 TRACK NG |
| EB SPZERO | Setpoint zero scale | R | Float | Expressed in engineering units |
| EB STNVL | Station value | R/W | Boolean<br>String 14 | 0 LOCAL<br>1 COMPUTER |
| EB STNMODE | Cascade/ratio or normal flag | R/W | Boolean<br>String 14 | 0 NORMAL<br>1 CASCADE/RATIO |

*Table A 15 Station Tag Attributes (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB PVQUAL | PV qua ty | R | Boo ean<br>Str ng 14 | 0 GOOD<br>1 BAD |
| EB STNTYPE | Station type | R | nteger<br>Str ng 32 | 0 BAS C W TH SET POINT<br>1 BASIC W THOUT SET PO NT<br>2 BAS C W TH B AS<br>3 RAT O<br>4 CASCADE |

**NOTES**
1 nd cates the operations that can be performed on the tag read (R) write (W) and subst tute (S)
2 Not supported by the O S 40 Series conso e or Conductor VMS workstation nstead use the fo ow ng attr butes EB AUTOMAN EB ST"LVL EB ST"MODE EB CMP"RS"S

*Table A 16 TEXT Tag Attributes*

| Attribute | Descript on | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB BLINK | Text blink enabled flag | R/S | Boolean | 0 no b nk<br>1 bl nk |
| EB COLOR | Text color | R/S | nteger<br>Str ng 12 | 0 BLACK<br>1 WHITE<br>2 RED<br>3 GREEN<br>4 BLUE<br>5 CYAN<br>6 MAGEN"A<br>7 YELLOW<br>8 ORANGE<br>9 YELLOW GREEN<br>10 GREEN CYAN<br>11 CYAN BLUE<br>12 BLUE MAGENTA<br>13 MAGENTA RED<br>14 DARK GRAY<br>15 LIGHT GRAY |
| EB MSG | Message number | R/S | Integer<br>String 80 | 0 9999<br>A conf gured text str r g stored n the supported system s assoc ated w th each message number |

**NOTE**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 17 TEXTSTR Tag Attributes*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMSUP | Enab e/d sable alarm suppress on | R/W | Boo ean | 0 d sabled<br>1 enabled |
| EB AUTOMAN | Auto/manua mode | R/W | Boo ean<br>String 6 | 0 MANUAL<br>1 AUTO |
| EB BLKLEN | Module defined text ength | R | nteger | 1 80 |
| EB CNTRLPER | Control perm ss ble flag | R | Boo ean<br>Str ng 14 | 0 DISABLED<br>1 ENAB_ED |
| EB MODELOCK | Mode nter ock | R | Boo ean<br>String 14 | 0 - NO MODE LOCK<br>1 MODE LOCK |
| EB O SLEN | Console text ength | R | nteger | 1 80 |
| EB PDTECHO | Echo flag | R | Boolean<br>Str ng 14 | Except on report from b ock<br>Exception report from contro |

*Table A 17 TEXTSTR Tag Attributes (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB QUALOVR | Qual ty overndden f ag | R/W[2] | Boo ean<br>Str ng 14 | 0 NORMAL<br>1 OVERR DDEN |
| EB RCVLEN | Actua ength | R | nteger | Up to 80 characters |
| EB REALM | Re-a arm togg e | R | Boo ean<br>Str ng 14 | 0 ZERO<br>1 ONE |
| EB REDTAG | Red tagged | R | Boo ean<br>Str ng 14 | 0 NORMAL<br>1 – RED TAGGED |
| EB SEQNUM | Sequence number of th s excep tion report | R | Integer | 0 255 |
| EB STRLOCK | Str ng interlock enabled flag | R | Boo ean<br>Str ng 14 | 0 UNLOCKED<br>1 LOCKED |
| EB TRUNC | Text string truncat on f ag | R | Boo ean<br>String 14 | 0 NO TRUNCAT ON<br>1 REMOTE TRUNCAT ON |
| EB TXTSTR | Text string data | R/W/S | Str ng 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 mited by EB O SLEN

## *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<br>1 acknow edged |
| C ADPKB | Symbo c state | R | Integer | 0 8 where 0 a keyboards |
| EB ADPLAMP | A arm state defin t on | R | Integer | 0 - 32 where 0 none |
| EB ADPANEL | Zero state descriptor | R | Integer | 0 4 where 0 none |
| EB ALMGRP | Alarm group | R | nteger<br>String 2 | 0 - 99<br>100 S<br>101 D |
| EB AUTOINHB | A arm auto nh bited flag | R | Boolean | 0 no<br>1 – a arm auto nh bited |
| EB BADPR O | Alarm priority for bad qua ty a arm | R | Integer,<br>String 8 | 1 - 8, 1 has the h ghest pr ority<br>1 DIRE<br>2 CRUCIAL<br>3 CRIT CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |
| EB GBLACK | Transmit g oba acknow edge f ag | R | Boo ean | 1 g oba acknow edge |
| EB GRP NHB | nh b ted by a arm group flag | R | Boo ean | 1 transm t globa acknow edge |
| EB NHBONST | Inh bit on state flag | R | Boo ean | True for d g ta state reporting tags only |

*Table A 18 All Tag Type Attributes (Secondary) (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB NHBST | A arm nh b t ng a arm status or d g ta state | R | nteger Str ng 2 | For EB NHBONST true 0 3<br><br>For EB NHBONST fa se H 2H 3H L 2L, 3L HD LD HR LR A or blank[2] |
| EB NHBTAG | Alarm inhibit tag | R | String 14 | Blank padded tag name or all b anks (d sab ed)[3] |
| EB MAN NHBPER | Manua nh bit ng of alarms enab ed flag | R | Boo ean | 1 manual nh b t of a arms enab ed |
| EB NODELIS | Node st number for g oba a arm management | R | nteger | 0 node st bu t by he conso e at start up from the f rs 32 N90 STA tags<br>1 4 conf gured node list |
| EB PRTALM | Pr nt a arm messages flag | R | Boo ean | 1 pr nt a arm messages |
| EB PRTOPACT | Pr nt operator act on messages f ag | R | Boo ean | 1 pr nt operator messages |
| EB RTN | Return to normal flag | R | Boo ean | 1 return to norma |
| EB RTNCMT | A arm comment for return to nor ma state | R | Str ng 64 | 1 64 characters |
| EB RTNPR O | Alarm pr or ty for return to norma a arm | R | nteger Str ng 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUCIAL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 INFORM |
| EB SAVALM | Save alarm messages f ag | R | Boo ean | 1 save a arm messages |
| EB SAVOPACT | Save operator action messages f ag | R | Boo ean | 1 save operator ac on messages |
| EB SECGRP | Security group | R | nteger | 1 16 |
| EB SECLVL | Secur ty level | R | nteger | 1 16 |
| EB SPECRCVD | Exception report specificat ons have been received flag | R | Boo ean | 1 ~ exception report specif cations rece ved |

**NOTES**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)
2 The va id va ues a so depend on the tag type of the a arm nh b t 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 DECD G | F oating po nt d g ts | R | nteger | 0 4 |
| EB H CMT | A arm comment for h gh a arm | R | Str ng 64 | 0 64 characters |
| EB H PR O | A arm pnor ty for h gh a arm | R | nteger Str ng 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUCIAL<br>3 CR TICAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |

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

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB LOCMT | A arm comment for low a arm flag | R | Str ng 64 | 0 64 characters |
| EB LOPR O | *A arm pr onty for low a arm* | R | nteger<br>String 8 | 1 8 1 has *the h ghest prior ty*<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |

**NOTE**
1 nd cates the operatio ns that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 20 DAANALG Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMTYPE | A arm type | R | nteger<br>String 8 | STANDARD h gh to ow<br>F XED 3 high to 3 ow<br>VAR ABLE dynam c h gh and low |
| EB DECDIG | Symbol c state | R | Integer | 0 - 4 |
| EB HI2CMT | Alarm comment for h gh h gh *alarm* | R | Str ng 64 | 0 64 characters |
| EB HI2PR O | A arm prior ty of h gh h gh a arm | R | Integer<br>Str ng 8 | 1 - 8, 1 has the highest pnor ty<br>1 DIRE<br>2 CRUCIAL<br>3 CRIT CAL<br>4 ALERT<br>5 – WARN NG<br>6 ADVISE<br>7 NOT FY<br>8 – INFORM |
| EB HI3CMT | Alarm comment for h gh h gh high alarm | R | Str ng 64 | 0 64 characters |
| EB *HI2PRIO* | *Alarm pnority of h gh h gh high* a arm | R | *Integer*<br>String 8 | 1 8 1 has *the highest prior ty*<br>1 D RE<br>2 CRUC AL<br>3 CR TICAL<br>4 ALERT<br>5 – WARN NG<br>6 – ADVISE<br>7 NOTIFY<br>8 – INFORM |
| EB HICMT | Alarm comment for high alarm | R | Str ng 64 | 0 - 64 characters |
| EB HIPR O | Alarm priority of high a arm | R | Integer<br>Str ng 8 | 1 8, 1 has the highest pr or ty<br>1 D RE<br>2 – CRUC AL<br>3 CR TICAL<br>4 – ALERT<br>5 WARN NG<br>6 ADV'SE<br>7 NOT FY<br>8 NFORM |
| EB LO2CMT | A arm comment for ow ow alarm | R | Strng 64 | 0 64 characters |

*Table A 20 DAANALG Attributes (Secondary) (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB LO2PR O | A arm pr or ty of ow low a arm | R | nteger<br>Str ng 8 | 1 8 1 has the h gh est pr or ty<br>1 D RE<br>2 CRUC AL<br>3 CR TICAL<br>4 ALERT<br>5 WARN NG<br>6 ADVISE<br>7 NOTIFY<br>8 NFORM |
| EB LO3CMT | A arm comment for ow low low a arm | R | Str ng 64 | 0 64 characters |
| EB LO3PR O | A arm pr ority of low low ow a arm | R | Integer<br>String 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARNING<br>6 ADVISE<br>7 NOT FY<br>8 INFORM |
| EB LOCMT | A arm comment for low alarm | R | Str ng 64 | 0 64 characters |
| EB LOPR O | A arm pr onty of ow alarm | R | Integer<br>Str ng 8 | 1 8 1 has the h ghest pnority<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |

**NOTE**
1 nd cates the operat ons that can be performed cn the tag read (R) wr te (W) and subst tute (S)

*Table A 21 DAANG Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB DECD G | F oat ng po nt d g ts | R | Integer | 0 4 |
| EB H 2CMT | A arm comment for h gh high alarm | R | String 64 | 0 64 characters |
| EB H 2PRIO | Alarm pr or ty of h gh h gh a arm | R | nteger | 1 8, 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUC AL<br>3 CRIT CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOTIFY<br>8 NFORM |
| EB H 3CMT | A arm comment for h gh h gh h gh a arm | 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 alarm | R | Integer String 8 | 1 8 1 has the highest priority<br>1 DIRE<br>2 CRUCIAL<br>3 CRITICAL<br>4 ALERT<br>5 WARNING<br>6 ADVISE<br>7 NOTIFY<br>8 INFORM |
| EB HICMT | Alarm comment for high alarm | R | String 64 | 0 64 characters |
| EB HPRIO | Alarm priority of high alarm | R | Integer String 8 | 1 8 1 has the highest priority<br>1 DIRE<br>2 CRUCIAL<br>3 CRITICAL<br>4 ALERT<br>5 WARNING<br>6 ADVISE<br>7 NOTIFY<br>8 INFORM |
| EB HDEVCMT | 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<br>1 DIRE<br>2 - CRUCIAL<br>3 - CRITICAL<br>4 ALERT<br>5 - WARNING<br>6 - ADVISE<br>7 NOTIFY<br>8 INFORM |
| EB HRATECMT | 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<br>1 DIRE<br>2 CRUCIAL<br>3 CRITICAL<br>4 - ALERT<br>5 - WARNING<br>6 ADVISE<br>7 NOTIFY<br>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<br>1 DIRE<br>2 CRUCIAL<br>3 CRITICAL<br>4 - ALERT<br>5 WARNING<br>6 ADVISE<br>7 NOTIFY<br>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 LO3PR O | A arm pr or ty of ow low ow a arm | R | nteger Str ng 8 | 1 8 1 has the h ghest prior ty<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |
| EB LOCMT | A arm comment for ow a arm | R | Str ng 64 | 0 64 characters |
| EB LOPR O | Alarm pr or ty of ow a arm | R | nteger Strng 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |
| EB LODEVCMT | A arm comment for ow dev at on a arm | R | Strng 64 | 0 64 characters |
| EB LODEVPR O | Alarm prior ty of ow dev at on a arm | R | nteger Str ng 8 | 1 8 1 has the highest prior ty<br>1 D RE<br>2 CRUCIAL<br>3 CRIT CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |
| EB LORATECMT | A arm comment for ow rate a arm | R | Str ng 64 | 0 64 characters |
| EB LORATEPR O | Alarm prior ty of ow rate a arm | R | Integer String 8 | 1 - 8, 1 has the h ghest pr or ty<br>1 – D RE<br>2 CRUC AL<br>3 CRIT CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 – NOT FY<br>8 NFORM |

**NOTE**
1 nd cates the operat ons 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 d g tal a arm | R | String 64 | 0 64 characters |
| EB ALMPR O | Alarm pnonty for d ç ta a arm | R | nteger String 8 | 1 8, 1 has the h ghest pr or ty<br>1 DIRE<br>2 CRUC'AL<br>3 CRIT CAL<br>4 ALERT<br>5 WARN NG<br>6 ADVISE<br>7 NOT FY<br>8 NFORM |
| EB PRTSTCHG | Pr nt state change messages f ag | R | Boo ean | 1 pr nt 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 | Boo ean | 1 save state change messages |

**NOTE**
1 nd cates the operations that can be performed on the tag read (R) wr te (W) and subst tute S)

*Table A 23 Device Driver (DD) Tag Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMCMT | A arm comment for d g tal alarm | R | Str ng 64 | 0 64 characters |
| EB A MPR O | A arm priority for dig ta a arm | R | nteger<br>Str ng 8 | 1 - 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUC AL<br>3 CR TICAL<br>4 ALERT<br>5 ~ WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |
| EB PRTSTCHG | Pr nt state change messages f ag | R | Boolean | 1 pr nt state changes |
| EB SAVSTCHG | Alarm pr or ty of h gh h gh a arm | R | Boolean | 1 save state changes |

**NOTE**
1 nd cates the operations that can be performed on the tag read (R) wnte (W) and substitute (S)

*Table A 24 DEVSTAT Tag Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMCMT | A arm comment for DEVSTAT[2] a arm | R | Stnng 64 | 0 64 characters |
| EB ALMPR O | A arm prior ty for DEVSTAT alarm | R | nteger<br>String 8 | 1 - 8, 1 has the h ghest pr onty<br>1 ~ D RE<br>2 CRUCIAL<br>3 CRITICAL<br>4 ALERT<br>5 WARNING<br>6 ADV SE<br>7 NOTIFY<br>8 INFORM |
| EB DEVTYPE | Device type | R | Integer<br>Strng 16 | 0 undefined<br>1 printer |
| EB PRTNUM | Printer number | R | nteger | 1 4 |
| EB PRTTYPE | Pnnter type | R | nteger<br>Str ng 10 | 0 ANSI<br>1 ~ ANS color<br>2 BM<br>3 IBM co or<br>5 ~ ANSI h gh<br>6 ~ BM B&W 24<br>7 BM col24 |

**NOTES**
1 ndicates the operations that can be performed on the tag read (R) write (W) and subst tute (S)
2 Only appl cable on the O S 40 Series conso e and Conductor VMS workstat on for determ n ng the pnnter type The dev ce status for a pnnter s not ma nta ned via the device status tag

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

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMCMT | A arm comment for d gital 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 ALMPR O | Alarm priority for digital alarm | R | Integer<br>Str ng 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 INFORM |
| EB PRTSTCHG | Pr nt state change messages f ag | R | Boo ean | pr nt state change messages |
| EB SAVSTCHG | Save state change message flag | R | Boo ean | 1 save state change messages |

**NOTE**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 26 MSDD Tag Attributes (Secondary)*

| Attribute | Descript on | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMCMT | A arm comment for d g ta a arm | R | Str ng 64 | 0 64 characters |
| EB ALMPRIO | Alarm pr ority for d gital alarm | R | Integer<br>String 8 | 1 8 1 has the h ghest priority<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARNING<br>6 ADV SE<br>7 NOT FY<br>8 INFORM |
| EB PRTSTCHG | Pr nt state change messages f ag | R | Boo ean | 1 print state change messages |
| EB SAVSTCHG | A arm pr or ty of h g h gh a arm save state change message flag | R | Boo ean | 1 save state change messages |

**NOTE**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 27 N90STA Tag Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMCMT | Alarm comment for N90STA a arm | R | String 64 | 0 64 characters |
| EB ALMPR O | A arm prior ty for N90STA a arm | R | nteger<br>String 8 | 1 8 1 has the h ghest priority<br>1 D RE<br>2 CRUCIAL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADVISE<br>7 NOTIFY<br>8 NFORM |
| EB MODCFGACK | Module configurat on error acknow edged f ag | R | Boo ean | 1 modu e configurat on error acknow edged |
| EB MODERRACK | Modu e errors ex st acknow edged f ag | R | Boo ean | 1 modu e errors ex st acknow edged |
| EB MODMODEACK | Module mode error acknow edged f ag | R | Boolean | 1 module mode error acknowl edged |
| EB MODRTN | Commun cat ons module return to norma flag | R | Boo ean | 1 commun cat ons modu e return to norma |

*Table A 27 N90STA Tag Attributes (Secondary) (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB NCFGACK | Node configurat on error acknow edged flag | R | Boo ean | 1 node conf gurat on error acknow edged |
| EB NCOMMACK | Node commun cat on error acknow edged flag | R | Boo ean | 1 node commun cat on error acknow edged |
| EB NMODERRACK | Modu e errors n a node ex st acknowledged flag | R | Boo ean | 1 modu e errors n a node ex st acknow edged |
| EB NOFFL NEACK | Node off- ne error acknow edged f ag | R | Boo ean | 1 Node off ne error acknow edged |
| EB NSYSACK | Node system status error acknow edged flag | R | Boo ean | 1 node system status error acknow edged |
| EB PCUINHB | PCU nhibit f ag | R | Boolean | 1 PCU nh b t |
| EB PCUSCANSTS | PCU scan off flag | R | Boo ean | 1 PCU scan off |
| EB XM TACK | Transm t globa acknowledge enab ed flag | R | Boo ean | 1 transmit g oba acknow edge enab ed |
| EB XM TSIL | Transm t g oba s ence enabled flag | R | Boolean | 1 transm t g oba s ence enab ed |

**NOTE**
1 nd cates the operation s that can be performed on the tag read (R) wr te (W) and substitute (S)

*Table A 28 RCM Tag Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMCMT | A arm comment for d g ta a arm | R | String 64 | 0 64 characters |
| EB ALMPR O | Alarm priority for dig tal a arm | R | nteger, Str ng 8 | 1 8, 1 has the h ghest pr ority<br>1 D RE<br>2 CRUC AL<br>3 CR T CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |
| EB PRTSTCHG | Print state change messages f ag | R | Boolean | 1 – pr nt state change messages |
| EB SAVSTCHG | A arm prior ty of h gh h gh a arm save state change message flag | R | Boo ean | 1 save state change messages |

**NOTE**
1 ndicates 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 | Str ng 64 | 0 - 64 characters |
| EB ALMPR O | Alarm priority for dig tal alarm | R | nteger Str ng 8 | 1 8, 1 has the h ghest priority<br>1 – D RE<br>2 CRUC AL<br>3 – CR TICAL<br>4 A ERT<br>5 WARN NG<br>6 ADV SE<br>7 NOTIFY<br>8 NFORM |
| EB PRTSTCHG | Pr nt state change messages f ag | R | Boo ean | 1 – pr nt state change messages |

*Table A 29 RMCB Tag Attributes (Secondary) (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB SAVSTCHG | A arm prior ty of h gh h gh a arm save state change m essage flag | R | Boo ean | 1 save state change messages |
| EB TXTSET | Text set number | R | nteger | 0 100 0 nd cates no text set |

**NOTE**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 30 RMSC Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB DECDIG | Floating point digits | R | nteger | 0-4 |

**NOTE**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 31 Station Tag Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB DECD G | F oat ng point d g ts | R | nteger | 0 - 4 |
| EB HICMT | Alarm comment for high alarm | R | String 64 | 0 64 characters |
| EB H DEVCMT | A arm comment for h gh deviat on alarm | R | Str ng 64 | 0 64 characters |
| EB H DEVPR O | Alarm priority of h gh dev at on a arm | R | nteger String 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUCIAL<br>3 CRIT CAL<br>4 – ALERT<br>5 WARN NG<br>6 ADVISE<br>7 NOT FY<br>8 NFORM |
| EB HIPR O | A arm priority for high alarm | R | Integer Str ng 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUCIAL<br>3 CRITICAL<br>4 ALERT<br>5 WARNING<br>6 ADVISE<br>7 NOT FY<br>8 – NFORM |
| EB LOCMT | Alarm comment for low alarm | R | String 64 | 0 64 characters |
| EB_LODEVCMT | A arm comment for ow dev at on a arm | R | String 64 | 0 64 characters |
| EB LODEVPR O | A arm priority for ow dev ation alarm | R | nteger String 8 | 1 - 8 1 has the h ghest pr or ty<br>1 DIRE<br>2 CRUCIAL<br>3 CR TICAL<br>4 ALERT<br>5 WARNING<br>6 ADVISE<br>7 NOT FY<br>8 NFORM |

*Table A 31 Station Tag Attributes (Secondary) (continued)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB LOPR O | Alarm priority for ow a arm | R | nteger<br>String 8 | 1 8 1 has the h ghest pr or ty<br>1 D RE<br>2 CRUC AL<br>3 CRIT CAL<br>4 ALERT<br>5 WARN NG<br>6 ADV SE<br>7 NOT FY<br>8 NFORM |
| EB TUNEBLK | Block number of P D used for tun ng | R | Integer | 11 9998 |

**NOTE**
1 nd cates the operat ons that can be performed on the tag read (R) wr te (W) and subst tut a (S)

*Table A 32 TEXT Tag Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB PRTSTCHG | Print state change messages f ag | R | Boo ean | 1 – print state change messages |
| EB SAVSTCHG | Alarm priority of h gh h gh a arm save state change message flag | R | Boolean | 1 save state change messages |

**NOTE**
1 nd cates the operations that can be performed on the tag read (R) wr te (W) and subst tute (S)

*Table A 33 TEXTSTR Tag Attributes (Secondary)*

| Attribute | Description | R/W/S[1] | Data Type | Value |
|---|---|---|---|---|
| EB ALMCMT | A arm comment for TEXTSTR a arm | R | String 64 | 0 64 characters |
| EB ALMPR O | A arm priority for TEXTSTR alarm | R | nteger,<br>String 8 | 1 8 1 has the h ghest pr onty<br>1 DIRE<br>2 CRUC AL<br>3 CR T CAL<br>4 – ALERT<br>5 WARN NG<br>6 ADVISE<br>7 NOTIFY<br>8 – INFORM |

**NOTE**
1 nd cates the operations that can be performed on the tag read (R) write (W) and substitute (S)

# Index

# Index (cont nued)