

**Previsio™**

White Paper

# **OPC Interface Description**

**June 2013**

Prepared by:



**Previsio Inc.®**  
Managing Critical Microsoft® Technology

### Printed in Canada

This document is subject to continuous improvement, and as such is subject to change without notice. Feedback or inquiries regarding this manual are welcome. Contact Previsio via [www.previsio.com](http://www.previsio.com).

---

**Proprietary Notice:** This manual, and related software, contain proprietary information which represents trade secrets of Previsio Inc. and may not be copied or disclosed except as provided in the license with Previsio Inc. Use of the information in this manual, and related software, for the reverse engineering of OPsCon,™ or for the development or manufacture of similar software is prohibited. The information in this manual is subject to change without notice and should not be construed as a commitment by Previsio Inc. Previsio Inc. assumes no responsibility for any errors that may be in this document.

Copyright © 1998 - 2013 Previsio Inc.® All rights reserved.

Unauthorized reproduction is a violation of Previsio Inc. copyright.

### Trademarks

Previsio® and OPsCon™ are trademarks of Previsio Inc.

Bailey®, Network 90®, Net 90®, Infi 90®, Conductor NT, and SYMPHONY™ are registered trademarks of Eltag Bailey Process Automation N.V.

Microsoft®, Windows NT®, and Microsoft SQL Server™, are trademarks or registered trademarks of the Microsoft Corporation.

All other brand or product names are trademarks or registered trademarks of their respective holders.

### Notice

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

# Table of Contents

<b>1</b>	<b>OPC Server for Bailey Net 90 and Infi 90 .....</b>	<b>2</b>
1.1	Introduction .....	2
1.2	OPC Overview .....	2
1.3	Install and Set-Up of OPC Server.....	2
1.4	OPC DA Browse Tree.....	3
1.5	Supported Function Codes .....	3
1.6	Reading and Writing to the OPC DA Server .....	4
1.7	Interpreting the Tables In this Manual .....	7
<b>2.</b>	<b>OPC Items at OPC Server Root Level.....</b>	<b>9</b>
<b>3.</b>	<b>OPC Items at Communications Channel (CIU) Level .....</b>	<b>10</b>
<b>4.</b>	<b>OPC Items at Process Control Unit (PCU) Level .....</b>	<b>11</b>
<b>5.</b>	<b>OPC Items at Module Level .....</b>	<b>12</b>
<b>6.</b>	<b>OPC Items at Tag (Block or Function Code) Level .....</b>	<b>14</b>
6.1	Common OPC DA Items for All Tags.....	14
6.2	Analog Exception Report (FC 30) .....	16
6.3	Digital Exception Report (FC 45) .....	17
6.4	Remote Control Memory (FC 62).....	18
6.5	Remote Manual Set Constant (FC 68) .....	19
6.6	Control Station (FC 80, 21, 22, 23) .....	20
6.7	Device Driver (FC 123).....	21
6.8	Multi-State Device Driver (FC 129).....	23
6.9	Remote Motor Control Block (FC 136).....	24
6.10	Text Selector (FC 151) .....	25
6.11	User Defined Data Export (FC 194) .....	26
6.12	Data Acquisition Analog (FC 177) .....	27
6.13	(Harmony) Analog In/Channel (FC 222).....	29
6.14	(Harmony) Analog Out/Channel (FC 223) .....	30
6.15	(Harmony) Digital In/Channel (FC 224).....	31
6.16	(Harmony) Digital Out/Channel (FC 225) .....	32
6.17	Analog and Digital Output Blocks.....	33
<b>7</b>	<b>Event State Attributes .....</b>	<b>36</b>
7.1	Operation Overview.....	37
7.2	EventRaw#1 Details .....	40
7.3	EventState Details.....	41
7.4	Configuration via BaileyOPCServer.INI File .....	42
<b>8.</b>	<b>OPC Alarm and Event Server Interface .....</b>	<b>43</b>
8.1	Event Types .....	43
8.2	Common Event Header Information .....	43
8.3	Supported Sub-Conditions for Events .....	44
8.4	Event Attributes.....	45
8.5	Message Header Information.....	45
8.6	Message Attributes.....	45
	<b>Appendix A – Error Codes.....</b>	<b>46</b>

# 1 OPC Server for Bailey Net 90 and Infi 90

---

## 1.1 Introduction

This document provides a description of the OPC™ Server interface. This is an OPC™ compliant interface, which provides access to the Bailey Net 90 and Infi 90 DCS systems.

This interface has the following features:

- Compliant with OPC™ Data Access (DA) Specification 2.0
- Compliant with OPC Alarms & Events (A&E) Specification 1.02
- Permits read and write access to Net 90 and Infi 90 DCS
- Supports module status functions
- Requires minimal configuration after installation

---

**NOTE:** This OPC interface supports additional features that are not documented in this manual. These features are provided for the support of specific OPC clients. Refer to the ActiTune ActiveX tuning control manuals, the CLU manuals, and to documentation for other OPC clients for further information.

---

---

## 1.2 OPC Overview

OLE for Process Control (OPC™) is an interface technology designed to allow Windows based applications from different vendors to communicate via a standard interface. You may find out more about OPC at [www.opcfoundation.org](http://www.opcfoundation.org).

---

## 1.3 Install and Set-Up of OPC Server

Before you attempt to read or write data via the OPC interface, it is assumed that the OPC Server has been properly installed, configured and set-up. These operations are beyond the scope of this manual, which just describes the OPC interface.

To install and set-up the OPC Server, to make the OPC interface ready for data acquisition, refer to the appropriate installation manual.

## 1.4 OPC DA Browse Tree

The accompanying diagram illustrates the OPC browse tree provided with the OPC Server. The levels in this browse tree are:

**Root** – This level is for the driver itself, and provides attributes for the driver.

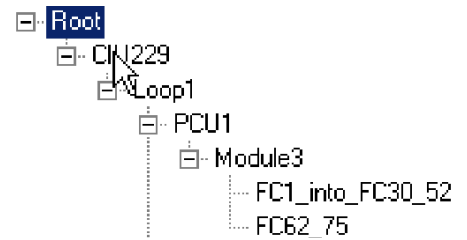
**CIU (Channel)** – This level is for each CIU (or communication channel) to the DCS. This OPC Server can support multiple simultaneous connections via serial and SCSI interfaces. Refer to other documentation for supported interfaces

**Loop** – This level is for the DCS “Loop” or data highway. No attributes are available at this level

**PCU** – This level is for the DCS PCU (Process Control Unit), which is usually a cabinet.

**Module (Device)** – This level is for the module, and provides attributes for Module status. The OPC Server supports module status functions. In addition, there are OPC client applications available to provide additional module functionality.

**Tag** – This level is for the tag, which connects to a block within a module. Section 1.5 contains a list of the supported function codes.



## 1.5 Supported Function Codes

Data Type	Block Name	Function Codes Supported
ANALOG	Analog	FC 30, FC 222, FC 223
DIGITAL	Digital	FC 45, FC 224, FC 225
RCM	Remote Control Memory	FC 62
RMSC	Remote Manual Set Constant	FC 68
STATION	Station	FC 80, 21, 22, 23
DD	Device Driver	FC 123
MSDD	Multi-State Device Driver	FC 129
RMCB	Remote Motor Control	FC 136
TEXT	Text Block	FC 151
TEXTSTR	User Defined Data Export	FC194
DAANG	DAANG	FC 177
XAI (ANALOG ENH/IN)	(Harmony) Analog In/Channel	FC 222
XAO (ANALOG ENH/OUT)	(Harmony) Analog Out/Channel	FC 223
XDI (DIGITAL ENH/IN)	(Harmony) Digital In/Channel	FC 224
XDO (DIGITAL ENH/OUT)	(Harmony) Digital Out/Channel	FC 225
ANALOGO	Analog Input Loop	FC 30 to FC 26
DIGITALO	Digital Input Loop	FC 45 to FC 42

Non exception generating function codes are supported via additional OPC client functions that are not described in this document.

---

## 1.6 Reading and Writing to the OPC DA Server

### Root Level

None of the Items at the Root level support Write operations. Read from Root Items using the following format:

**<Server>.< Item>**

Where:

**<Server>** the base server address for OPC Server

**<Item>** the specific OPC Item taken from the detailed tables.

### CIU Level

None of the objects at the CIU level support Write operations. Read from CIU items using the following format:

**<Server><CIU>.< Item>**

Where:

**<Server>** the base server address for OPC Server

**<CIU>** the name assigned for the CIU (alphanumeric string)

**<Item>** the specific OPC Item taken from the detailed tables.

### Loop Level

There are no read/write Items at the Loop Level

### PCU Level

None of the objects at the PCU level support Write operations. Read from PCU items using the following format:

**<Server><CIU>.<Loop>.<PCU>.< Item>**

Where:

**<Server>** the base server address for OPC Server

**<CIU>** the name assigned for the CIU (alphanumeric string)

**<Loop>** the name assigned for the Loop (alphanumeric string)

**<PCU>** the name assigned for the PCU (alphanumeric string)

**<Item>** the specific OPC Item taken from the detailed tables.

## Module Level

None of the objects at the Module level support Write operations. Read from Module items using the following format:

Read/write to specific Tags using the following format:

**<Server><CIU>.<Loop>.<PCU>.<Module>.< Item>**

Where:

**<Server>** the base server address for OPC Server

**<CIU>** the name assigned for the CIU (alphanumeric string)

**<Loop>** the name assigned for the Loop (alphanumeric string)

**<PCU>** the name assigned for the PCU (alphanumeric string)

**<Module>** the name assigned for the Module (alphanumeric string)

**<Item>** the specific OPC Item taken from the detailed tables.

## Tag Level (Long Form)

Read/write to specific Tags using the following format:

**<Server><CIU>.<Loop>.<PCU>.<Module>.<Block>.< Item>**

Where:

**<Server>** the base server address for OPC Server

**<CIU>** the name assigned for the CIU (alphanumeric string)

**<Loop>** the name assigned for the Loop (alphanumeric string)

**<PCU>** the name assigned for the PCU (alphanumeric string)

**<Module>** the name assigned for the Module (alphanumeric string)

**<Block>** the tag name assigned for the Block (alphanumeric string) using the Configurator.

**<Item>** the specific OPC Item taken from the detailed tables.

## Tag Level (Short Form)

The OPC Server also supports a short form for Tag addressing.

Read/write to specific Tags using the following (short form) format:

**<Server>.<Block>.< Item>**

Where:

**<Server>** the base server address for OPC Server

**<Block>** the tag name assigned for the Block (alphanumeric string) using the Configurator.

**<Item>** the specific OPC Item taken from the detailed tables.

## Reading an Arbitrary Block Address

You may also use the OPC server to read an arbitrary block address. This does not need to be exception tag.

The correct syntax to read an arbitrary block address is as follows:

**<CIU>@<LOOP>:<PCU>:<MODULE>:<BLOCK>**

or

**<CIU>#<LOOP>:<PCU>:<MODULE>:<BLOCK>**

Examples include:

CIUnn@1:2:3:4

CIUnn#1:2:3:4

Where:

- CIU = The CIU address as shown in the Configurator system tree
- LOOP = Loop Address (integer)
- PCU = PCU address (integer)
- MODULE = module address (integer)
- BLOCK = block address (integer)

The SCAN interval is defined as an attribute of the OPC data group defined by the OPC client. Note however that:

- If you are using the OPC server for acquisition of exception based data as well (e.g. console) then you will find that the period varies as the volume of exception data varies.
- You will find that there is a lower limit to the period (in seconds) that you will be able to achieve. You will need to experiment to see where that lower limit is. It is doubtful that you will be able to acquire data at reliably fixed intervals below 1 second.
- The period that you are able to achieve will depend on how many values you are trying to acquire. The more values you attempt to acquire, the lower the period will be for each value.



## 1.7 Interpreting the Tables In this Manual

This section provides a guide to interpreting the detailed tables in this manual. OPC Items are listed in tables of the following form:

OPC Items Available at .....						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation

### ***OPC Item***

The OPC Item Name

### ***Item Description***

A brief name or description of this Item

### ***T (Best Data Type)***

Best Data Type (T) is selected from the following options:

- B - Boolean
- D – Double (Float64)
- F – Float32
- L – Int32
- S – Int16
- T – String
- N – Numeric (any of types Int16, Int32, Float32 and Double may be used as native data type)

You may access most OPC items using any of these data types. However, for each OPC item only one of these data types is considered the best data type. You will achieve best results if you use this data type for access.

### ***Q (OPC Item Status and Quality)***

Per OPC Data Access standard v2.0, each OPC item is provided with a matching status variable.

This status variable, as seen at the Previs OPC Test Client interface, will normally have one of the following values:

- 00C0 HEX if all aspects of it's matching OPC Item are GOOD (i.e. no problem).
- 0000 HEX to indicate BAD.

The OPC Data Access standard provides for other possible values, to provide further fault information if available. These additional sub-status states are not supported. Refer to the OPC Data Access version 2.0 specification for further details.

The logic used to set GOOD or BAD status at the OPC interface provides for the following three situations:

- **AG – Always Good** – OPC Quality is always set to GOOD.
- **CM – Communications** – OPC Quality is set to BAD if communications is lost between the Bailey DCS and the OPC Server.
- **HW – Hardware** – OPC Quality is set to BAD if (a) communications is lost between the Bailey DCS and the OPC Server OR (b) the Bailey DCS Quality Flag (i.e. “Q”) for this tag is set to 1 (0 = OK; 1 = BAD).

Please note that the OPC Item Status is assigned individually for each individual OPC Item. Refer to the tables within this document to see how OPC Item Status is assigned for any given OPC Item (i.e. AG, CM, HW). Unless otherwise noted all OPC Items Status are of type CM.

### ***W (Read/Write Type)***

Defines whether the Item is Read Only (R), Write Only (W), or Read/Write (RW).

### ***Value***

This field provides the domain of values for the Item

### ***Interpretation***

This field provides additional interpretive documentation.

## 2. OPC Items at OPC Server Root Level

The accompanying table lists all OPC Items available at the ROOT level within the OPC Browse Tree. These OPC Items provide attributes of the Driver itself.

OPC Items Available at Server Root Level of Browse Tree						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Status Fields (Updated at Driver Status Change)</b>						
DrvSta	Driver Status	N	AG	R	Numeric	-2 = Shutting Down; -1 = Driver Application Just Started; 0 = Stopped 1 = Connecting; 2 = Normal Run State
HeartBeat	Incrementing counter to show "I am alive"	L	AG	R	Integer (0+)	Integer count starts at 0 at start of driver execution and increments once per second thereafter (treat as unsigned Int32)
<b>Internal Use Only Fields (Not for General use)</b>						
SrvBusy	Internal Use Only – Do not use					
FileList	Internal Use Only – Do not use					
ModuleFileList	Internal Use Only – Do not use					

### 3. OPC Items at Communications Channel (CIU) Level

The accompanying table lists all OPC Items available at the Communications Channel (CIU) Level within the OPC Browse Tree. These OPC Items provide attributes of the CIU Connection itself.

OPC Items Available at Communications Channel (CIU) Level of Browse Tree						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Status Fields (Updated at Driver Status Change)</b>						
Status	CIU Status	N	AG	R	Numeric	-2 = Shutting Down; -1 = Driver Application Just Started; 0 = Stopped 1 = Connecting; 2 = Normal Run State
<b>Performance Fields (Updated Continuously)</b>						
Tuning Parameter	DCS Communications Tuning Parameter	F	AG	RW	Numeric	For tuning DCS communications. Default 500 (recommended), Minimum 100, Maximum 1250. Increase(Decrease) to reduce(increase) communications overhead while slowing(speeding) responsiveness of exception traffic.
StatXmtTotal	# packets transmitted	F	AG	R	Numeric	Total count of packets transmitted from driver startup or last Configurator reset
StatXmtTotalRate	Packet Transmit Rate	F	AG	R	Numeric	Packet Transmit Rate, in packets per second, average over recent time
StatRcvTotal	# packets received	F	AG	R	Numeric	Total count of packets received from driver startup or last Configurator reset
StatRcvTotalRate	Packet Receive Rate	F	AG	R	Numeric	Packet Receive Rate, in packets per second, averaged over recent time
StatExceptionPacketsRate	Exception Packet Receive Rate	F	AG	R	Numeric	Exception Packet Receive Rate, in exception packets per second, averaged over recent time
StatExceptionRate	Exception Rate	F	AG	R	Numeric	Instantaneous rate, in exceptions per second, for the last packet received, measured as (number of exceptions in packet/last exception packet interval)
StatExceptionRateAverage	Exception Rate Average	F	AG	R	Numeric	Exception Receive rate, in exceptions per second, averaged over recent time
StatExceptionInPacket	# Exceptions in Packet	F	AG	R	Numeric	Number of exceptions in last exception packet received
StatLastInterval	Last Interval Time	F	AG	R	Numeric	Time, in milliseconds, between last two exception packets received
StatErrorTimeoutTotal	# of Timeouts	F	AG	R	Numeric	Total count of number of timeouts since last counter reset
StatRetryTotal	# of Retries	F	AG	R	Numeric	Total count of number of retries since last counter reset
StatErrorTotal	# of Errors	F	AG	R	Numeric	Total count of number of errors since last counter reset
StatErrorOverrunTotal	# of Overruns	F	AG	R	Numeric	Total count of number of overruns since last counter reset
StatLastError	Last error received	F	AG	R	0 to 255	If non 0 this may (and may not) be an error. Refer to Appendix A for details
<b>Internal Use Only Fields (Not for General use)</b>						
StatReadCountMax	Read Queue Size	F	AG	R	Numeric	Suggests problem if, and only if, >8 AND incrementing continuously
StatWriteCountMax	Write Queue Size	F	AG	R	Numeric	Suggests problem if, and only if, >8 AND incrementing continuously
Cmd	Internal Use Only – Do not use					
Property	Internal Use Only – Do not use					

## 4. OPC Items at Process Control Unit (PCU) Level

The accompanying table lists all OPC Items available at the Process Control Unit (PCU) Level within the OPC Browse Tree. These OPC Items provide attributes of the PCU itself.

OPC Items Available at Process Control Unit (PCU) Level of Browse Tree						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Status Fields (Updated at Driver Status Change)</b>						
Type	Node Type	T	AG	R		Node type string as entered at Configurator
<b>Internal Use Only Fields (Not for General use)</b>						
ErrorIndicators	Internal Use Only					

## 5. OPC Items at Module Level

The accompanying table lists all OPC Items available at the Module Level within the OPC Browse Tree. These OPC Items provide attributes of the Module itself.

OPC Items Available for Module Level of Browse Tree						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Items updated when Configurator Change is Made</b>						
Name	Module Name	T	AG	R	Text string	Module name
Description	Description of this module	T	AG	R	Text string	Module Description
HwAddress	Hardware Address for this module	T	AG	R	Text string	<CIUname>@<Loop>:<PCU>:<Module>
<b>Items Updated via DCS Exception only</b>						
ErrSum	Error Summary	S	CM	R	0, 1	0 = OK, 1 = Errors Present
Mode	Module operating Mode	T	CM	R	Integer AND/OR Text String	0 = CONFIGURE Mode; 1 = FAILED Mode; 2 = ERROR Mode; 3 = EXECUTE Mode; If used as an integer this returns an integer. If used as a string then it returns a string.
Q	Quality	S	CM	R	Integer AND/OR Text String	0 = <empty string> = Good Quality 1 = Q = Bad Quality If used as an integer this returns an integer. If used as a string then it returns a string.
SWRev	Revision number of module on-board firmware	T	CM	R	string	Provides firmware revision. Example "A7"
FTX	First time in execute	S	CM	R	Boolean	1 = YES; 0 = NO
BOK	Backup module or memory failed	S	CM	R	Boolean	1 = Failed; 0 = OK
RIO	Remote I/O status	S	CM	R	Boolean	1 = BAD; 0 = OK
LIO	Local I/O status	S	CM	R	Boolean	1 = BAD; 0 = OK
CAL	Calibration Quality status	S	CM	R	Boolean	1 = BAD; 0 = OK
AIE	Auto Initialization Input Status	S	CM	R	Boolean	1 = SET; 0 = RESET
EAI	ROM Contains Default Configuration	S	CM	R	Boolean	1 = YES; 0 = NO
STA	Summary Station Status	S	CM	R	Boolean	1 = SET; 0 = RESET

OPC Items Available for Module Level of Browse Tree						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
Type	Module type	T	CM	R	Integer AND/OR Text String	0=REZVD(REZERVED) 1=PIM01 2=CTM01 3=AMM01 4=AOM01 5=COM02(COM02/03/04) 6=CLC01 7=LMM01(NLMM01) 8=LIM01(NLIM01/02, NBIM01/INBIM02) 9=AMM02(NAMM02) 10=PCI01(INPCI01) 11=MFC01(IMMFC01/02) 12=LMM02(IMLMM02) 13=MPC01(IMMPC01) 14=BTM01(NLIM01/02, INBTM01) 15=LCM01(INLCM01) 16=LSM01(NLSM01/INPCT01) 17=GCM01(INGCM01) 18=" " 19=" " 20=" " 21=" "(EXTENDED) 22=IPT01,(EXTENDED+INIPT01) 23=" " 24=IPT02(EXTENDED+INIPT02) 25=SLC01(EXTENDED+SLC01) 26=" " 27=" " 28=" " 29=" " 30=LCM02(INLCM02) 31=LCM03(INLCM03) 32=MCP02(IST,ICT,PST,IIMCP02) 33=SBM01(SBM01) 34=SCM(SCM) 35=BCM(BCM INIT01/03) 36=MFP(IMMFP01/02/03) 37=NPM01(NPM01) 38=" " 39=NIT02(INIT02) 40=MCP(MCP) 41=" " 42=SEM01(SEM01)
If used as an integer this returns an integer. If used as a string then it returns a string.						
<b>Performance Fields (Updated Continuously)</b>						
StatXmtTotal	# packets transmitted	F	AG	R	Numeric	Total count of packets transmitted from startup or last Configurator reset
StatRcvTotal	# packets received	F	AG	R	Numeric	Total count of packets received from startup or last Configurator reset
StatExceptions	# exceptions received	F	AG	R	Numeric	Total count of exceptions, specifically for the Module and not it's tags, received since driver startup or last reset
StatLastInterval	Last Interval Time	F	AG	R	Numeric	Time, in milliseconds, between last two exception packets received
StatLastException	Time Since Last Exception	F	AG	R	Numeric	Time, in milliseconds, since last exception received (updated at defined update rate for this OPC group)
StatErrorTotal	# of Errors	F	AG	R	Numeric	Total count of number of errors since last counter reset
StatLastError	Last error received	F	AG	R	0 to 255	If non 0 this may (and may not) be an error. Refer to Appendix A for details
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						
<b>Internal Use Only Fields (Not for General use)</b>						
Status	Binary of Module Status Information	T	CM	R	16 bytes	Binary data describing module status. Recommend that this NOT be used ! If you need to view status, use one of the OPC clients provided for this purpose.
Pr	Binary array of Problem Reports	T	CM	R	Hex string	Binary data describing problem reports. Recommend this NOT be used ! If you need to view problem reports, use one of the OPC clients available.
Property	For Internal Use only (Do not Use)					

## 6. OPC Items at Tag (Block or Function Code) Level

### 6.1 Common OPC DA Items for All Tags

OPC Items Available and Common to all Exception Block Types (not ANALOGO or DIGITALO)						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Updated via Configurator Only</b>						
Tag_Name	Name of this tag	T	AG	R	Text string	Tag Name
Description	Description of this tag.	T	AG	R	Text string	Tag Description
HwAddress	Hardware address for the module containing this tag	T	AG	R	Text string	<CIUname>@<Loop>:<PCU>:<Module>.Block>
FC_Type	Function Code Type	T	AG	R	Text String	String selected from <ANALOG, DIGITAL, RCM, RMSC, STATION, DD, MSDD, RMCB, TEXT, DAANG, ANALOGO, DIGITALO> to indicate Function Code type per table in section 1.5 herein.
ALARM_PRTY	Alarm Priority	T	AG	R	Text string	String selected from {INFO, LOLO, LOW, MEDIUM, HIGH, HIHI,CRITICAL}
ALARM_AREA	Alarm area that this tag is assigned to	T	AG	R	Text string	Define alarm area that this tag belongs too
ALARM_EXT1	80 character string, usually used as alarm description	T	AG	R	Text string	User Defined
ALARM_EXT2	80 character string, usually used as alarm description	T	AG	R	Text string	User Defined
<b>Alarm Inhibit to A&amp;E Interface</b>						
Inhibit_CFG	Describes inhibit logic	T	AG	R	Text String	String " <i>Inhibit if &lt;source tag name&gt; = &lt;inhibit state&gt;</i> " where source tag name (define at configurator) is source of inhibit signal and inhibit state defines when inhibited (e.g. at OPEN, CLOSE, ALARM, ZERO, ONE, TWO, THREE)
InhibitValue	A&E interface inhibit state	T	AG	R	0,1	Alarms to A&E interface ARE (1) or ARE NOT (0) inhibited
<b>Performance Fields (Updated Continuously)</b>						
AlarmCounter	# of Alarms Received	F	AG	R	Numeric	Running count, from last driver start or reset, or alarm related state transitions.
StatExceptions	# exceptions received	F	AG	R	Numeric	Total count of exceptions, specifically for this Tag, received since driver startup or last reset
StatLastInterval	Last Interval Time	F	AG	R	Numeric	Time, in milliseconds, between last two exception packets received
StatLastException	Time Since Last Exception	F	AG	R	Numeric	Time, in milliseconds, since last exception received (updated at defined update rate for this OPC group)
StatErrorTotal	# of Errors	F	AG	R	Numeric	Total count of number of errors since last counter reset
StatLastError	Last error received	F	AG	R	0 to 255	If non 0 this may (and may not) be an error. Refer to Appendix A for details





## 6.2 Analog Exception Report (FC 30)

OPC Items Available for Analog Exception Report - FC30						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Tag Output Value	F	HW	R	Real	Read analog input value
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
ALM	Limit Alarm State	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Alarm, 2 = Hi Alarm
DA	Deviation Alarm State	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Dev Alarm, 2 = Hi Dev Alarm
RT	Red Tag indicator	S	CM	R	0,1	1 = Tagged, 0 = Not Tagged
SPT	Set Point Tracking Indicator	S	CM	R	0,1	0 = Not Tracking, 1 = Tracking
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>						
UNIT	Engineering Units Index	S	CM	R	Integer	e.g. Spec S2 for FC30 in engineering units. EU string is either (a) indexed from string array in BaileyOPCServer.INI via EUINDEX specification or (b) entered via Configurator.
ZERO	Zero	F	CM	R	Real	e.g. Spec S3 for FC30 in engineering units
SPAN	Span	F	CM	R	Real	e.g. Spec S4 for FC30 in engineering units
HALM	High alarm limit	F	CM	R	Real	e.g. Spec S5 for FC30 in engineering units
LOALM	Low alarm limit	F	CM	R	Real	e.g. Spec S6 for FC30 in engineering units
SUPD	Reserved (Do not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.3 Digital Exception Report (FC 45)

OPC Items Available for Digital Exception Report - FC45						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Tag Current Value	S	HW	R	0,1	Read digital input value
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
ALM	Limit Alarm State	S	CM	R	0,1	0 = Normal, 1 = Alarm
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>						
ALMSPEC	Alarm Specification	S	CM	R	0,1,2	e.g. Spec S2 for FC45 0 = Logic 0 Alarm; 1 = Logic 1 Alarm; 2 = No Alarm State Defined
SUPD	Reserved (Do Not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.4 Remote Control Memory (FC 62)

OPC Items Available for Remote Control Memory - FC62							
OPC Item	Item Description	T		W	Domain of Values		
					Value	Interpretation	
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>							
CV	Tag Output Value	S	HW	RW	Integer	Read	Read Tag Output 0 or 1
		S				Write	0 = Request Pulsed Reset 1 = Request Pulsed Set
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK	
ALM	Alarm State	S	CM	R	0,1	0 = Normal; 1 = Alarm	
RT	Red Tag Status	S	CM	R	0,1	1 = Tagged, 0 = Not Tagged	
LS	Logic Set Input Received	S	CM	R	0,1	0 = No, 1 = Yes	
SP	Set Permissive Input Received	S	CM	R	0,1	0 = No - set permissive input to NOT 1 1 = Yes - set permissive input to 1	
LR	Logic Reset Input Received	S	CM	R	0,1	0 = No, 1 = Yes	
OV	Override	S	CM	R	0,1	0 = NOT Override; 1 = Override	
FV	Feedback Value	S	CM	R	0,1	0 = No, 1 = Yes	
SC	Set Command Received	S	CM	R	0,1	0 = No, 1 = Yes	
RC	Reset Command Received	S	CM	R	0,1	0 = No, 1 = Yes	
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>							
TYPE	Type Parameter (Selected Display Type)	S	CM	R	Integer 0 to 255	e.g. Spec S8 for FC62 0 = Output Indicator, 1 = No output Indicator 2 = Output and Feedback Indicator 3 = Feedback Indicator Only	
SUPD	Reserved (Do Not use)						
<b>Event State Fields</b>							
Includes event state fields described in Section 7.							

## 6.5 Remote Manual Set Constant (FC 68)

OPC Items Available for Remote Manual Set Constant - FC68						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Tag Output Value	F	HW	RW	Real	Read field to get value; write to field to change value
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
ALM	Limit Alarm State	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Alarm, 2 = Hi Alarm
DA	Deviation Alarm State	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Dev Alarm, 2 = Hi Dev Alarm
RT	Red Tag Indicator	S	CM	R	0,1	1 = Tagged, 0 = Not Tagged
SPT	Set Point Tracking Indicator	S	CM	R	0,1	0 = Not Tracking, 1 = Tracking
<b>Function Code Specifications From DCS (Read Only - Updated at console startup and whenever specifications changed)</b>						
UNIT	Engineering Units Index	S	CM	R	Integer	e.g. Spec S1 for FC68 in engineering units. EU string is either (a) indexed from string array in BaileyOPCServer.INI via EUINDEX specification or (b) entered via Configurator.
ZERO	Engineering unit number Zero	F	CM	R	Real	Calculated from other values
SPAN	Engineering unit number Span	F	CM	R	Real	Calculated from other values
HIALM	High limit	F	CM	R	Real	e.g. Spec S2 for FC68 in engineering units
LOALM	Low limit	F	CM	R	Real	e.g. Spec S3 for FC68 in engineering units
SUPD	Reserved (Do Not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.6 Control Station (FC 80, 21, 22, 23)

OPC Items Available for Control Station - FC80, FC21, FC22, FC23							
OPC Item	Item Description	T	Q	W	Domain of Values		
					Value	Interpretation	
<b>Write ONLY Attribute for State Change Command</b>							
MD_CMD	Mode Command. WRITE issues command for mode change (functions the same as write function for attribute MD). READ returns last command sent to DCS (without actual feedback).	S	--	RW	Integer	0 - Go to Local-Manual (Console/Station-Manual) 1 - Go to Local Auto (Console/Station-Auto) 2 - Go to Local Cascade/Ratio (Control/Station - Cascade/Ratio) 3 - Go to Computer Manual 4 - Go to Computer Auto 5 - Go to Computer Cascade/Ratio 6 - Go to Local Level (Cascade/Station Level) 7 - Go to Computer Level 8 - Go to Computer Backup State 9 - Computer OK 10 - Go to Previous State	
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>							
MD	Mode	S	HW	RW	Integer	Read	0 = Manual, 1 = Automatic 2 = Cascade Ratio - Manual, 3 = Cascade/Ratio Automatic
						Write	0 - Go to Local-Manual (Console/Station-Manual) 1 - Go to Local Auto (Console/Station-Auto) 2 - Go to Local Cascade/Ratio (Control/Station - Cascade/Ratio) 3 - Go to Computer Manual 4 - Go to Computer Auto 5 - Go to Computer Cascade/Ratio 6 - Go to Local Level (Cascade/Station Level) 7 - Go to Computer Level 8 - Go to Computer Backup State 9 - Computer OK 10 - Go to Previous State
CV	Control Output	F	HW	RW	Real	Read	CO (Control Output )
						Write	Write Control Output when in Manual Mode
SP	Set Point	F	CM	RW	Real	Read/Write Set Point	
RI	Ratio Index	F	CM	RW	Real	Read/Write Ratio Index	
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK	
RT	Red Tag Indicator	S	CM	R	0,1	1 = One or more Station points Tagged, 0 = No Points Tagged	
ALM	Limit Alarm <sup>1</sup>	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Alarm, 2 = High Alarm	

<sup>1</sup> All station variables take the same limit alarm as that of the process variable.

OPC Items Available for Control Station - FC80, FC21, FC22, FC23						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
SPT	Set Point Tracking	S	CM	R	0,1	0 = Normal, 1 = Tracking
PV	Process Variable	F	HW	R	Real	Station Process Variable
DA	Deviation Alarm State	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Deviation Alarm, 2 = High Deviation Alarm
PVAS	Process Variable Alarm State	S	CM	R	Integer	0 = No Alarm, Not 0 = Alarm
PVQ	Process Variable Quality	S	AG	R	0,1	0 = OK, 1 = Calibration out of Range
OTS	Output Tracking State	S	CM	R	0,1	0 = Not Tracking, 1 = Tracking
AOS	Analog Output Status	S	CM	R	0,1	0 = OK, 1 = Digital Station Failure
SBM	Station Bypass Mode	S	CM	R	0,1	0 = Station Bypass Not Enabled, 1 = Station Bypass Enabled
MI	Mode Interlock	S	CM	R	0,1	0 = Mode Interlock Not set, 1 = Mode Interlock Set
CS	DCS Computer Status	S	CM	R	0,1	0 = Computer OK, 1 = Computer failed, station mode dependent on S17
SMD	Computer Cascade Status	S	CM	R	0,1,2,3	Aggregates STN Level and Cascade-Ratio / Normal 0 = Local, 1 = Cascade/Ratio Local 2 = Computer Level, 3 = Computer Cascade/Ratio
UNIT	Engineering Units Index	S	CM	R	Integer	Index to Engineering Units String. EU string is either (a) indexed from string array in BaileyOPCServer.INI via EUINDEX specification or (b) entered via Configurator.
<b>Function Code Specifications From DCS (Read Only - Updated at console startup and whenever specifications changed)</b>						
HIALM	High Alarm Setting	F	CM	R	Real	Spec S07 for FC80, FC21, FC22, FC23 in Engineering Units
LOALM	Low Alarm Setting	F	CM	R	Real	Spec S08 for FC80, FC21, FC22, FC23 in Engineering Units
DEVALM	Deviation Alarm Setting (PV/SP)	F	CM	R	Real	Spec S09 for FC80, FC21, FC22, FC23 in Engineering units
SPAN	PV and SP Span	F	CM	R	Real	Spec S10 for FC80, FC21, FC22, FC23 in Engineering Units
PVZERO	Process variable zero	F	CM	R	Real	Spec S11 for FC80, FC21, FC22, FC23 In Engineering Units
SPZERO	Set Point zero	F	CM	R	Real	Spec S14 for FC80, FC21, FC22, FC23
TYPE	Station Type	S	CM	R	1,2,4,8,16	Spec S23 for FC80 1 = Basic with Set Point, 2 = Ratio Index, 4 = Cascade 8 = Basic without Set Point, 16 = Basic with Bias
SUPD	Reserved (Do Not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.7 Device Driver (FC 123)

OPC Items Available for Device Driver - FC123						
OPC	Item Description	T	Q	W	Domain of Values	

					Value	Interpretation	
<b>Write ONLY Attribute for State Change Command</b>							
CV_CMD	Device Command. WRITE issues command for state change (functions the same as write function for attribute CV). READ returns last command sent to DCS (without actual feedback).	S	CM	RW	Integer	1 = Request Set Control Output Equal to 1 2 = Request Reset Control Output Equal to 0 4 = Request Manual Mode 8 = Request Automatic Mode	
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>							
CV	Tag output value	S	HW	RW	Integer	Read	Value of 0 or 1
						Write	1 = Request Set Control Output Equal to 1 2 = Request Reset Control Output Equal to 0 4 = Request Manual Mode 8 = Request Automatic Mode
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK	
ALM	Alarm state	S	CM	R	0,1	1 = Alarm, 0 = No Alarm	
FB1	Feedback 1	S	HW	R	0,1	Displays contents of FB1O and FB1C	
FB2	Feedback 2	S	HW	R	0,1	Displays contents of FB2O and FB2C	
FBS	Feedback Status	S	HW	R	0,1	0 = Good, 1 = Bad	
SO	Status Override	S	CM	R	0,1	0 = No, 1 = Yes (Override)	
SD	User defined state	S	CM	RW	Integer	Read/Write field for user defined state	
RT	Red Tag Status	S	CM	R	0,1	1 = Tagged, 0 = Not Tagged	
MD	Operating Mode	S	CM	R	0,1,2,3	0 = Auto, 1 = Remote, 2 = Manual, 3 = N/A	
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>							
TYPE	Device Driver Display Type	S	CM	R	0 to 255	e.g. Specification S10 for FC123 Usually set to 0 to select faceplate for this function code	
<b>Event State Fields</b>							
Includes event state fields described in Section 7.							



## 6.8 Multi-State Device Driver (FC 129)

OPC Items Available for Multi-State Device Driver - FC129							
OPC Item	Item Description	T	Q	W	Domain of Values		
					Value	Interpretation	
<b>Write ONLY Attribute for State Change Command</b>							
CV_CMD	Device Command. WRITE issues command for state change (functions the same as write function for attribute CV). READ returns last command sent to DCS (without actual feedback).	S	CM	RW	Integer	0 = Request change to State 0 1 = Request change to State 1 2 = Request change to State 2 3 = Request change to State 3 4 = Request Change to Manual Mode 8 = Request Change to Automatic Mode	
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>							
CV	Tag Output Value	S	HW	RW	Integer	Read	Good State 0 = Zero, 1 = Four, 2 = Eight, 3 = Twelve
						Write	0 = Request change to State 0 1 = Request change to State 1 2 = Request change to State 2 3 = Request change to State 3 4 = Request Change to Manual Mode 8 = Request Change to Automatic Mode
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK	
ALM	Alarm State	S	CM	R	0,1	0 = No Alarm, 1 = Alarm	
FB1	Input 1 Feedback State	S	HW	R	0,1	Input 1 Feedback State = 1 (0 = No, 1 = Yes)	
FB2	Input 2 Feedback State	S	HW	R	0,1	Input 2 Feedback State = 2 (0 = No, 1 = Yes)	
FB3	Input 3 Feedback State	S	HW	R	0,1	Input 3 Feedback State = 3 (0 = No, 1 = Yes)	
FB4	Input 4 Feedback State	S	HW	R	0,1	Input 4 Feedback State = 4 (0 = No, 1 = Yes)	
SO	Status Override	S	CM	R	0,1	0 = No Override; 1 = Override	
SD	User defined state	S	CM	RW	Integer	Read/Write field for user defined state	
MD	Operating Mode	S	HW	R	0,1	0 = Manual Mode, 1 = Automatic Mode	
GS	Good State Table	S	HW	R	0,1,2,3	Good State: 0 = Zero, 1 = Four, 2 = Eight, 3 = Twelve	
RS	Requested State	S	CM	R	0,1,2,3	Requested State: 0 = Zero, 1 = One, 2 = Two, 3 = Three	
RT	Red Tag Status	S	CM	R	0,1	1 = Tagged, 0 = Not Tagged	
OVR	Control Override	S	CM	R	0,1	0 = No Override, 1 = Override	
<b>Function Code Specifications From DCS (Read Only - Updated at console startup and whenever specifications changed)</b>							
TYPE	MSDD Display Type	S	CM	R	Integer	e.g. Spec S18 for FC129; Expect normal value will be 0	
SUPD	Reserved (Do Not Use)						
<b>Event State Fields</b>							
Includes event state fields described in Section 7.							

## 6.9 Remote Motor Control Block (FC 136)

OPC Items Available for Remote Motor Control Block - FC136							
OPC Item	Item Description	T	Q	W	Domain of Values		
					Value	Interpretation	
<b>Write ONLY Attribute for State Change Command</b>							
CV_CMD	Device Command. WRITE issues command for state change (functions the same as write function for attribute CV). READ returns last command sent to DCS (without actual feedback).	S	CM	RW	Integer	1 = Request Reset 2 = Request Set 128 = Acknowledge Fault	
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>							
CV	Tag Output Value	S	HW	RW	Integer	Read	0 = Stopped, 1 = Running
						Write	1 = Request Reset; 2 = Request Set 128 = Acknowledge Fault
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK	
ALM	Alarm Status	S	CM	R	0,1	0 = No Alarm, 1 = Alarm	
FB1	Feedback 1 State	S	HW	R	0,1	Provides current state of feedback value	
FB2	Feedback 2 State	S	HW	R	0,1	Provides current state of feedback value	
SOH	Status on Hold	S	CM	R	0,1	0 = No, 1 = Yes	
RT	Red Tag Status	S	CM	R	0,1	1 = Tagged, 0 = Not Tagged	
BS	Bad Start	S	HW	R	0,1	0 = No (i.e. Good Start), 1 = No (i.e. Bad Start)	
FLT	Fault	S	HW	R	0,1	0 = No (i.e. No Fault), 1 = Yes (i.e. Fault)	
SP1	Start Permissive 1 State	S	CM	R	0,1	0 = No, 1 = Yes	
SP2	Start Permissive 2 State	S	CM	R	0,1	0 = No, 1 = Yes	
ERR	Error Message Code for Bad Start & Fault Conditions	S	CM	R	0, 1, ..., 9	0 = No error; 2 = interlock 1 equals 0; 4 = interlock 3 equals 0; 6 = feedback 1 equals 0 while running; 8 = feedback 1 equals 1 while stopped;	1 = Stopped (stop input = 1) 3 = interlock 2 equals 0 5 = interlock 3 equals 0 7 = feedback 2 equals 0 while running 9 = feedback 2 equals 1 while stopped
MD	Reserved (Do Not Use)						
LE	Latch Error Message	S	CM	R		Latched version of ERR (holds error until new error reported)	
<b>Function Code Specifications From DCS (Read Only - Updated at console startup and whenever specifications changed)</b>							
TYPE	Display Type	S	CM	R	Integer	e.g. Spec S14 for FC136 Expect normal value will be 0	
SUPD	Reserved (Do Not Use)						
<b>Event State Fields</b>							
Includes event state fields described in Section 7.							

## 6.10 Text Selector (FC 151)

OPC Items Available for Text Selector - FC151						
OPC Item	Item Description	T	Q	W	Domain of Value	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Message Number	S	HW	R	String	CV provides numerical string ID. The client application must use this to index through a string array.
MessageString	The selected message in string form	T	AG	R	String	The message string, indexed by CV, drawn from the string array contained in File <b>BaileyOPCServer.INI</b> . See NOTE 1 below for details.
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
CB	Color and Blink Combined	S	CM	R	Integer 0 to 255	High order bit is the blink bit (0 = no blink, 1 = blink) Low order 7 bits are Bailey Color (Recommend use COL and BLNK instead)
COL	Color	S	CM	R	integer 0 to 127	Bailey Defined Color
BLNK	Blink	S	CM	R	0, 1	0 = no blink, 1 = blink
ALM	Alarm Status	S	CM	R		0 = No Alarm, 1 = Alarm
SUPD	Reserved (Do Not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7. However, these fields perform no function for this Block Type. DO NOT USE.						

NOTE 1: OPC Server installs **BaileyOPCServer.INI** file at time of installation. This file contains a section headed [FC151] which, at install time, provides a set of default text strings for FC151. This INI file section will need to be edited for each project using FC151 text block to contain the project specific strings required. See INI file for details re editing this section.

## 6.11 User Defined Data Export (FC 194)

OPC Items Available for User Defined Data Export FC 194						
OPC Item	Item Description	T	Q	W	Domain of Value	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV <sup>2</sup>	Text String from HW	T	HW	RW	String	Text value (Read / Write) of string length [COUNT] characters
Q	Quality	N	AG	R	Integer 0,1	1 = Bad, 0 = OK
AUTO_MANUAL	Auto / manual	N	CM	R/W	Integer 0,1	0 = Manual, text can be changed by console 1 = Auto, text is changed by bailey block logic / program
COUNT	String length	N	CM	R	Integer 0 to 255	String length of the current text received from Bailey
SEQN	Sequence number	N	CM	R	Integer 0 to 255	This number is incremented by Bailey each time a new value is issued
ALM_LVL	Alarm Status / Level	N	CM	R	Integer 0,1,2,3	0 = No Alarm, 1 = HI alarm, 2 = HI2 alarm, 3 = HI3 alarm
TEXT_LOCK	Text lock status	N	CM	R	Integer 0,1	0 = Not Locked (block will accept text string command), 1 = Locked
MODE_LOCK	Mode lock status	N	CM	R	Integer 0,1	0 = Not Locked (block will accept mode), 1 = Locked (not accept command)
SUPD	Reserved (Do Not Use)					
<b>Function Code Specifications From DCS (Read Only - Updated at console startup and whenever specifications changed)</b>						
<b>Event State Fields</b>						
Includes event state fields described in Section 7. However, these fields perform no function for this Block Type. DO NOT USE.						

<sup>2</sup> Do not use CV to control Alarm Inhibit functions.

## 6.12 Data Acquisition Analog (FC 177)

OPC Items Available for DAANG – FC177										
OPC Item	Item Description	T	Q	W	Domain of Values					
					Value	Interpretation				
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>										
CV	Tag Current Value	F	HW	RW	Real	Read/Write analog value				
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK				
ALM	Limit Alarm State	S	CM	R	0,1, 2	0 = No Alarm, 1 = Low Alarm High Alarm				
ALLV	Alarm Level	S	CM	R	0,1,2,3	0 = Level 1, 1 = Undefined, 2 = Level 2, 2 = Level 3				
XS	Extended status Changed	S	CM	R	0,1	0 = No, 1 = Yes				
RT	Red Tag Indicator	S	CM	R	0,1	1 = Tagged, 0 = Not Tagged				
AMCAL	A/M mode + Value Calculated	S	CM	RW	0,1,2	<table border="1"> <tr> <td>Read</td> <td>0 = MANUAL, 1 = AUTO, 2 = CALCULATED</td> </tr> <tr> <td>Write</td> <td>                     0 = Request Manual Mode                      1 = Request Automatic (Input) Mode                      2 = Request Automatic (Calculated) Mode                      3 = Request Alarm Suppression                      4 = Request No Alarm Suppression                      5 = Request to Place On-Scan                      6 = Request to Place Off-Scan                      7 = Request Immediate Exception Report                 </td> </tr> </table>	Read	0 = MANUAL, 1 = AUTO, 2 = CALCULATED	Write	0 = Request Manual Mode 1 = Request Automatic (Input) Mode 2 = Request Automatic (Calculated) Mode 3 = Request Alarm Suppression 4 = Request No Alarm Suppression 5 = Request to Place On-Scan 6 = Request to Place Off-Scan 7 = Request Immediate Exception Report
Read	0 = MANUAL, 1 = AUTO, 2 = CALCULATED									
Write	0 = Request Manual Mode 1 = Request Automatic (Input) Mode 2 = Request Automatic (Calculated) Mode 3 = Request Alarm Suppression 4 = Request No Alarm Suppression 5 = Request to Place On-Scan 6 = Request to Place Off-Scan 7 = Request Immediate Exception Report									
PIS	Permit Input	S	CM	R	0,1	0 = No, 1 = Yes				
CE	Constraints Enabled	S	CM	R	0,1	0 = No, 1 = Yes				
HF	Hardware Failure/Bad Input Quality	S	CM	R	0,1	0 = No, 1 = Yes				
MLA	Multilevel Alarming	S	CM	R	0,1	0 = No, 1 = Yes				
OR	Suspect or Out of Range	S	CM	R	0,1	0 = OK, 1 = Suspect or Out of Range				
LIM	Limited	S	CM	R	0,1	0 = No, 1 = Yes				
QO	Quality Override Active	S	CM	R	0,1	0 = No, 1 = yes				
SS	No Report or Off Scan	S	CM	R	0,1	0 = No, 1 = Yes				
DA	Deviation Alarm	S	CM	R	0,1,2	0 = No Deviation Alarm, 1 = Low Deviation, 2 = High Deviation				
RA	Rate Alarm	S	CM	R	0,1,2	0 = No Rate Alarm, 1 = Low Rate Alarm, 2 = High Rate Alarm				
VA	Variable Alarms	S	CM	R	0,1	0 = No, 1 = Yes				
ASUP	Alarm Suppression Indication	S	CM	R	0,1	0 = Disabled, 1 = Enabled				
REA	Alarm in Re-Alarm Condition	S	CM	R	0,1	0 = No, 1 = Yes				

OPC Items Available for DAANG – FC177						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
NHL	Next Higher Limit	F	CM	R	Real	Next higher alarm limit
NLL	Next Lower Limit	F	CM	R	Real	Next Lower alarm limit
<b>Function Code Specifications From DCS (Read Only - Updated at console startup and whenever specifications changed)</b>						
UNIT	Engineering Units	S	CM	R	Integer	Spec S6 for FC177 - Engineering Unit Identifier. EU string is either (a) indexed from string array in BaileyOPCServer.INI via EUINDEX specification or (b) entered via Configurator.
HIALM	High Alarm setting	F	CM	R	Real	Spec S24 for FC177 - High Alarm
LOALM	Low Alarm Setting	F	CM	R	Real	Spec S25 for FC177 - Low Alarm
UVAL	User Value	F	CM	R	Real	Spec S13 for FC177 - Engineering Unit Selected
HILIM	High Limit	F	CM	R	Real	Spec S1 for FC177 - High Display Reference
CNTR	Center	F	CM	R	Real	Spec S2 for FC177 - Center Display Reference
LOLIM	Low Limit	F	CM	R	Real	Spec S3 for FC177 - Low Display Reference
TYPE	Type	S	CM	R	Integer	Not applicable or not used
SUPD	Reserved (Do Not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.13 (Harmony) Analog In/Channel (FC 222)

OPC Items Available for (Harmony) Analog In/Channel - FC 222						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Tag Output Value	F	HW	R	Real	Read analog input value
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
ALM	Limit Alarm State	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Alarm, 2 = Hi Alarm
SINH	Status Inhibit	S	CM	R	0,1	0 = Not Inhibited, 1 = Inhibited
SIM	Simulation Enabled	S	CM	R	0,1	0 = Simulation NOT enabled; 1 = Simulation Enabled
OVR	Value Override	S	CM	R	0,1	0 = No Override, 1 = Override
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>						
CHLABEL	Harmony Channel Label	T	CM	R	String	Channel label if defined at function block
EGULOW	Engineering Units Low Value	F	CM	R	Real	e.g. Spec S3 for FC222 in engineering units
EGUHIGH	Engineering Units High Value	F	CM	R	Real	e.g. Spec S4 for FC222 in engineering units
HIALM	High alarm limit	F	CM	R	Real	e.g. Spec S6 for FC222 in engineering units
LOALM	Low alarm limit	F	CM	R	Real	e.g. Spec S7 for FC222 in engineering units
SUPD	Reserved (Do not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.14 (Harmony) Analog Out/Channel (FC 223)

OPC Items Available for (Harmony) Analog Out/Channel - FC 223						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Tag Output Value	F	HW	R	Real	Read analog input value
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
ALM	Limit Alarm State	S	CM	R	0,1,2	0 = No Alarm, 1 = Low Alarm, 2 = Hi Alarm
SINH	Status Inhibit	S	CM	R	0,1	0 = Not Inhibited, 1 = Inhibited
SIM	Simulation Enabled	S	CM	R	0,1	0 = Simulation NOT enabled; 1 = Simulation Enabled
OVR	Value Override	S	CM	R	0,1	0 = No Override, 1 = Override
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>						
CHLABEL	Harmony Channel Label	T	CM	R	String	Channel label if defined at function block
EGULOW	Engineering Units Low Value	F	CM	R	Real	e.g. Spec S3 for FC223 in engineering units
EGUHIGH	Engineering Units High Value	F	CM	R	Real	e.g. Spec S4 for FC223 in engineering units
HIALM	High alarm limit	F	CM	R	Real	e.g. Spec S7 for FC223 in engineering units
LOALM	Low alarm limit	F	CM	R	Real	e.g. Spec S8 for FC223 in engineering units
SUPD	Reserved (Do not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						



## 6.15 (Harmony) Digital In/Channel (FC 224)

OPC Items Available for (Harmony) Digital In/Channel - FC 224						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Tag Current Value	S	HW	R	0,1	Read digital input value
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
ALM	Limit Alarm State	S	CM	R	0,1	0 = Normal, 1 = Alarm
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>						
CHLABEL	Harmony Channel Label	T	CM	R	String	Channel label if defined at function block
ALMSPEC	Alarm Specification	S	CM	R	0,1,2	e.g. Spec S2 for FC224 0 = Logic 0 Alarm; 1 = Logic 1 Alarm; 2 = No Alarm State Defined
SUPD	Reserved (Do Not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.16 (Harmony) Digital Out/Channel (FC 225)

OPC Items Available for (Harmony) Digital Out/Channel - FC 225						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Tag Current Value	S	HW	R	0,1	Read digital input value
Q	Quality	S	AG	R	0,1	1 = Bad, 0 = OK
ALM	Limit Alarm State	S	CM	R	0,1	0 = Normal, 1 = Alarm
<b>Function Code Specifications From DCS (Read Only – Updated at console startup and whenever specifications changed)</b>						
CHLABEL	Harmony Channel Label	T	CM	R	String	Channel label if defined at function block
ALMSPEC	Alarm Specification	S	CM	R	0,1,2	e.g. Spec S3 for FC225 0 = Logic 0 Alarm; 1 = Logic 1 Alarm; 2 = No Alarm State Defined
SUPD	Reserved (Do Not Use)					
<b>Event State Fields</b>						
Includes event state fields described in Section 7.						

## 6.17 Analog and Digital Output Blocks

This section describes the Analog Output block type (ANALOGO), and the Digital Output block type (DIGITALO).

ANALOGO and DIGITALO are the opposite direction of the analog and digital inputs. The blocks get configured within the control module of the CIU and are used to pass analog and digital values into the DCS. These are available *in addition to* other means of passing analog and digital values through other tag types for other function codes.

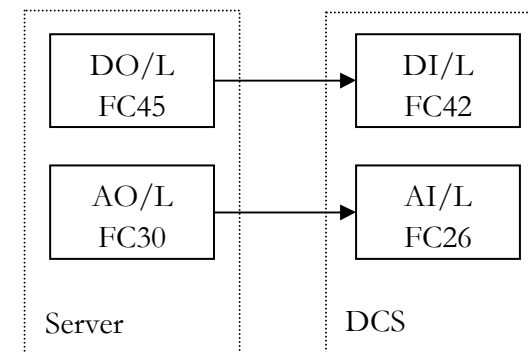
Within the Bailey DCS system, the ANALOGO will appear to be a standard Analog Exception Report (Function Code 30) and the DIGITALO will appear to be a standard Digital Exception Report (Function Code 45). The only difference is that the output of these blocks is controlled from within the OPC Client, to provide the ability to write analog and digital values to the DCS.

The user of these blocks can control the current output value, the tag or data quality and the alarm status of the tag. The outputs are intended for applications of the following type:

- Bridge applications where the OPC Server transfers analog or digital points to or from the DCS from another system.
- Supervisory control applications.
- Extending the analog and digital I/O of the Bailey system by adding I/O, to the OPC Server computer, with the proper OPC Client or other suitable driver.

These blocks function as DO/L and AO/L blocks within the Bailey DCS structure as shown in the accompanying diagram.

This functionality is also available via the OPC Server.



### 6.17.1 Analog Output Block (ANALOGO)

OPC Items Available for Analog Exception Report - FC30						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
CV	Current Analog Value	F	HW	RW <sup>1</sup>		Floating point variable
Q	Quality	S	AG	RW <sup>1</sup>		0: GOOD 1: BAD (NOTE: Defaults to BAD until first value written to this field)
ALM	Limit Alarm	S	CM	RW <sup>1</sup>		0: NO ALARM 1: LOW ALARM 2: HIGH ALARM
DA	Deviation Alarm	S	CM	RW <sup>1</sup>		0: NO ALARM 1: LOW ALARM 2: HIGH ALARM
RT	Red TAG Indicator	S	CM	RW		0 = NO RED TAG; 1 = RED TAG
SPT	Set Point Tracking Indicator	S	CM	RW		0 = TRACKING; 1 = NOT TRACKING
UNIT	Engineering units index	S	CM	RW		Engineering Unit Index (usually set to 0 unless Engineering Unit Index in use)
ZERO	Zero	F	CM	R		Set in Configurator
SPAN	Span	F	CM	R		Set in Configurator
HIALM	High Alarm Limit	F	CM	R		Set in Configurator
LOALM	Low Alarm Limit	F	CM	R		Set in Configurator
<b>Event State Fields</b>						
Includes event state fields described in Section 7. However, these fields perform no function for this Block Type. DO NOT USE.						

Note 1: Write to this field causes exception report containing current settings to be sent to DCS.

Note 2: To configure an ANALOGO for a CIU module the general steps required are:

1. Determine module address (i.e. Loop:PCU:Module) for CIU module (e.g. from Configurator).
2. Create an ANALOGO block within the OPC Server using the Configurator, at the Loop:PCU:Module (for CIU) address determined above. NOTE that the point index (i.e. tag index) and Block Address must be the same for this ANALOGO tag. (e.g. tag index = block address = 700 )
3. Create Function Code 26 block (e.g. using CLU) inside the module where you wish to receive the ANALOGO value. Specifications S1, S2 and S3 define the CIU and block address for the ANALOGO block from which you wish to receive data
4. To confirm that the ANALOGO value is being received at the target location, connect a FC30 block to the output of configured FC26 block and add the FC30 ANALOG block to the OPC Server so that you can see the feedback of the ANALOGO value you have written.

### 6.17.2 Digital Output Block (DIGITALO)

OPC Items Available for Digital Output Block (DIGITALO)						
OPC Item	Item Description	T	Q	W	Domain of Values	
					Value	Interpretation
<b>Function Code Fields Connected Directly to DCS (Updated upon Exception and, for read/write fields, when changed by Operator)</b>						
CV	Current Digital State	S	HW	RW <sup>1</sup>		0: OPEN 1: CLOSE
Q	Quality	S	AG	RW <sup>1</sup>		0: GOOD 1: BAD (NOTE: Defaults to BAD until first value written to this field)
ALM	Limit Alarm	S	CM	RW <sup>1</sup>		0: NO 1: ALARM
ALMSPEC	Alarm specification	S	CM	RW <sup>1</sup>		Set in Configurator: 0 = Logic 0 Alarm; 1 = Logic 1 Alarm; 2 = No Alarm State Defined
<b>Event State Fields</b>						
Includes event state fields described in Section 7. However, these fields perform no function for this Block Type. DO NOT USE.						

Note 1: Write to this field causes exception report containing current settings to be sent to DCS.

Note 2: Configure DIGITALO in a similar manner to configuring the ANALOGO.

## 7 Event State Attributes

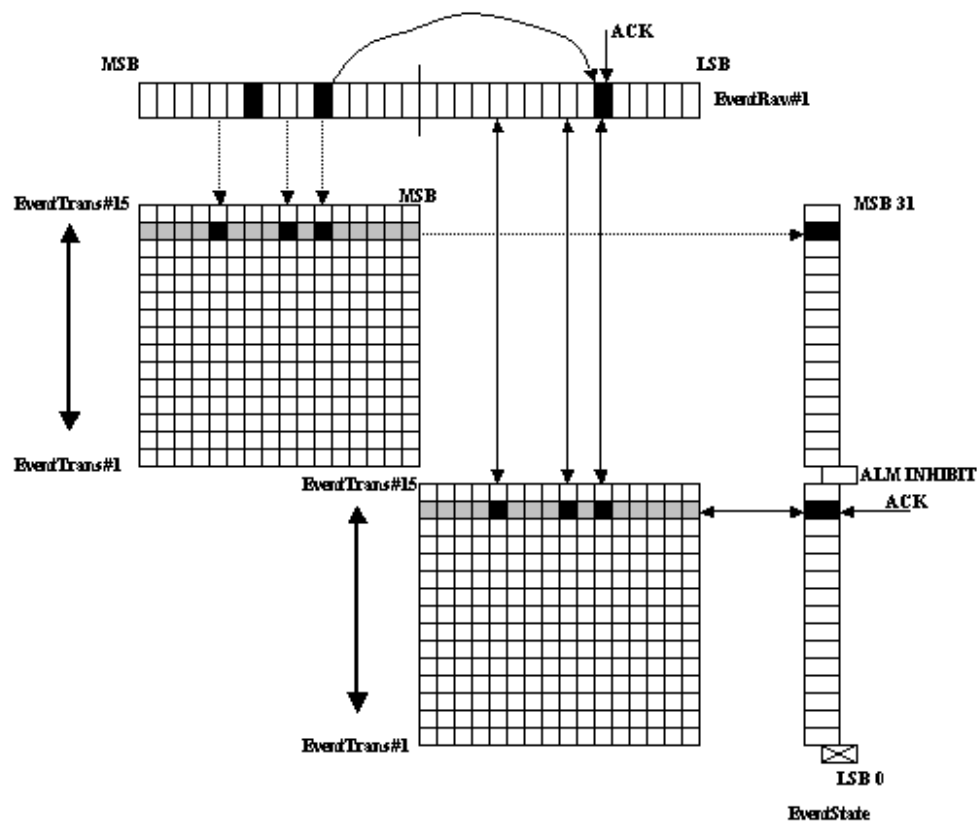
**NOTE:** This section describes the Event State attributes. These attributes have been added to support specific customers that require a high degree of control over specific alarm information. Many installations will not require use of the Event State information.

The following table lists the event state OPC Items that are common to all Module and Tag objects. The Event State features have been included in the OPC Server to support a re-mapping of Bailey Alarms into a generic alarm state format.

OPC Event State Attributes For Tags and Modules				
OPC Item	Item Description	W	Domain of Values	
			Value	Interpretation
<b>Attributes Updated ONLY at OPC Server startup via Read from File BaileyOPCServer.INI</b>				
EventTrans#1	EventTrans#1 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#2	EventTrans#2 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#3	EventTrans#3 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#4	EventTrans#4 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#5	EventTrans#5 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#6	EventTrans#6 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#7	EventTrans#7 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#8	EventTrans#8 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#9	EventTrans#9 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#10	EventTrans#10 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#11	EventTrans#11 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#12	EventTrans#12 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#13	EventTrans#13 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#14	EventTrans#14 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
EventTrans#15	EventTrans#15 bit map defined in INI file		2 bytes	Refer to description of BaileyOPCServer.INI file
<b>Attributes Updated at EITHER Exception from DCS OR at WRITE operation from OPC Client</b>				
EventRaw#1	Raw alarm and alarm ACK flags	√	32 bits	Raw alarm and alarm ACK flags/states (Refer to details below).
EventState	EventRaw#1 processed with EventTrans mapping masks	√	32 bits	Alarm and Alarm ACK states mapped by EventTrans Mapping Table (Refer to details below).

## 7.1 Operation Overview

The following diagram illustrates the manner in which EventRaw#1, EventTrans, and EventState work together.



**EventRaw#1 operations:**

The initial state of EventRaw#1 is 0x0000FFFF (HEX). This means “no alarms active and all alarms ACKed”.

The initial state of EventState is also 0x0000FFFF (HEX).

When an exception packet is received from the DCS, the corresponding bits in the upper 16 bits of EventRaw#1 are set (to 1) to indicate an anomalous state, and the corresponding bits in the lower 16 bits of EventRaw#1 are cleared (to 0) to indicate an UNACK Alarm. To ACK an alarm write 1 to the appropriate bit position in EventRaw#1.

**EventTrans Operations**

The BaileyOPCServer.INI file provides a number of 16 bit masks that may be used to select the EventRaw#1 bit positions that may contribute to setting an Alarm in EventState.

**EventState operations:**

The upper 15 bits of EventState are set as  $\langle \text{EventRaw\#1} \mid \text{Upper16bits} \rangle \text{AND} \langle \text{appropriate EventTrans mask} \rangle$ .

The lower 15 bits of EventState are set as  $\langle \text{EventRaw\#1} \mid \text{Lower16bits} \rangle \text{AND} \langle \text{appropriate EventTrans mask} \rangle$ .

Note that EventState bit 0 is not used, and EventState bit 16 is used solely to enable and disable alarms for this tag (This setting persists when the driver is stopped and started, and upon computer restart).

The values in Bit 1 through Bit 15 of EventState should be interpreted as 0 = UNACK Alarm (within the group defined by EventTrans mask) and 1 = NO UNACK Alarm.

To ACK an alarm write 1 to the appropriate bit position in EventState. This will also have the effect of generating an Alarm ACK for the corresponding alarm bit positions in EventRaw#1.

**Alarm acknowledgment:**

Note that alarms can be acknowledged via write to EITHER EventRaw#1 OR to EventState.

If the ACK is written to EventState, then the OPC server will attempt to clear all the Raw alarms bits where there is a “1” in the EventTrans mask.



**Loop alarm acknowledgment (currently applies to Net 90 Plant Loop only):**

Note that alarms can be acknowledged to the loop via write to EITHER EventRaw#1 OR to EventState. The acknowledge event is sent to all configured computer interfaces on the same loop. The destination computer interfaces receive the message and OPC Servers perform acknowledgement for all the alarm types. This limitation is caused by the format of communications protocols. The sender acknowledges alarms locally upon receiving of an OK response from the CIU for the sent message. There is no feedback from the destination node. The safe delivery of the Alarm ACK messages the system relies on the Bailey Loop communications protocols.

The feature can be enabled or disabled via configuration. At channel level there is a flag to enable/disable receive of Alarm ACK messages from other nodes. At tag level there is a flag to enable/disable sending of Alarm ACK messages to other nodes.

For this feature to operate both flags need to be enabled. Individual tags can have the Send Alarm ACK disabled. If the feature is enabled Receive Alarm ACK cannot be individually disabled for each tag.

## 7.2 EventRaw#1 Details

EventRaw#1 is a 32 bit unsigned long integer. The most significant 16 bits of EventRaw#1 contain alarm, event or status information received from the DCS via exception packets. The least significant 16 bits contain either Unacknowledged Alarm (bit = 0) or No Unacknowledged Alarm (bit = 1) in a 16 bit array that is mapped equivalent to the upper 16 bits. The following table describes EventRaw#1 mapping details.

<b>EventRaw#1 Composition</b>				
Bit Number		Assigned to This Bit	Description	Domain of Values for Upper 16 Bits
Upper	Lower			
<b>Applicable to All Tag Types</b>				
16	0	T/V	Tracking Or Digital Value	1 = Tracking (or Digital Value)
17	1	RT	Red Tag	1 = Red Tag flag set
18	2	C	Calibration	1 = Out of range
19	3	DL	Deviation Low	1 = Low Deviation State
20	4	DH	Deviation High	1 = High Deviation State
21	5	LA	Low Alarm	1 = Low Alarm Is Set
22	6	HA	High Alarm	1 = High Alarm Is Set
23	7	Q	Bad Quality	1 = Bad quality
<b>Applicable to Station Tag Type Only (FC 80, 21, 22, 23)</b>				
26	10	BYP	Bypassed	1 = In Bypass State
27	11	OT	Output Tracking	1 = In Output Tracking Mode
28	12	MI	Manual Interlock	1 = In Manual Interlock
29	13	DSF	Digital Station Failure	1 = Failure
30	14	COK	Computer OK	1 = Not OK
<b>Applicable to Device Driver (FC 123) and Multi-State Device Driver (FC 129)</b>				
24	8	FS	Feedback status bad	1 = Feedback status is bad
<b>Applicable to DAANG Tag Type Only (FC177)</b>				
24	8	F/Q	Hardware fail/Bad Quality input	1 = Bad quality or hardware failed
25	9	RL	Rate Low	1 = Alarm Set
26	10	RH	Rate High	1 = Alarm Set
27	11	LO2	Low alarm level 2	1 = Alarm Set
28	12	LO3	Low alarm level 3	1 = Alarm Set
29	13	HI2	High alarm level 2	1 = Alarm Set
30	14	HI3	High alarm level 3	1 = Alarm Set
<b>Applicable to TEXTSTR Tag Type Only (FC194)</b>				

EventRaw#1 Composition				
Bit Number		Assigned to This Bit	Description	Domain of Values for Upper 16 Bits
Upper	Lower			
16	0	AUTO_MANUAL	AUTO/MANUAL	1 = AUTO, 0 = MANUAL
17	1			1 = Alarm Set
18	2			1 = Alarm Set
19	3			1 = Alarm Set
20	4			1 = Alarm Set
21	5	ALM_LVL0	ALM_LEVEL is a 2 bit field	0 = NORMAL,1 = ALARM
22	6	ALM_LVL1		0 = NORMAL,1 = ALARM
23	7	Q	Bad quality	1 = BADQ, 0 = OK

### 7.3 EventState Details

The accompanying table provides further details regarding the EventState OPC attribute field.

EventState Composition		
Bit	Writable?	EventTrans Mask used for this bit (Use the EventTrans Record for this Object Type)
<b>Applicable to All Tag Types</b>		
0		This bit is not used
1	YES	EventTrans#1
2	YES	EventTrans#2
3	YES	EventTrans#3
••••		
13	YES	EventTrans#13
14	YES	EventTrans#14
15	YES	EventTrans#15
16	YES	Write 0 to Enable Alarms; Write 1 to Inhibit Alarms (Default is Alarm Enable State taken from Configurator) (Not affected)
17	NO	EventTrans#1
18	NO	EventTrans#2
19	NO	EventTrans#3
••••		

EventState Composition		
Bit	Writable?	EventTrans Mask used for this bit (Use the EventTrans Record for this Object Type)
29	NO	EventTrans#13
30	NO	EventTrans#14
31	NO	EventTrans#15

## 7.4 Configuration via BaileyOPCServer.INI File

The accompanying figure illustrates the format of the [CIUDRV] section of the BaileyOPCServer.INI file<sup>3</sup>. This file is loaded by the OPS Server at startup (only) to define the mapping from alarm and event states in the OPC Server EventState attribute.

Each line in the [CIUDRV] section of the BaileyOPCServer.INI file defines a mask that is used to map state information between EventRaw#1 and EventState.

Note that:

- The first bit in each line in the BaileyOPCServer.INI file maps to the MSB (most significant bit) of the upper/lower 16 bit segments of EventRaw#1 and Event State.

```
almANALOG_EventTrans1=0000000000000001
almANALOG_EventTrans2=0000000000000010
...
almANALOG_EventTrans14=0100000000000000
almANALOG_EventTrans15=1000000000000000
almDD_EventTrans1=0000000000000001
almDD_EventTrans2=0000000000000010
...
almDD_EventTrans14=0000000000000000
almDD_EventTrans15=0000000000000000
almDIGITAL_EventTrans1=0000000000000001
almDIGITAL_EventTrans2=0000000000000010
...
almDIGITAL_EventTrans14=0000000000000000
almDIGITAL_EventTrans15=0000000000000000
almMSDD_EventTrans1=0000000000000000
almMSDD_EventTrans2=0000000000000000
...
almMSDD_EventTrans14=0000000000000000
almMSDD_EventTrans15=0000000000000000
almRCM_EventTrans1=0000000000000000
almRCM_EventTrans2=0000000000000000
...
almRCM_EventTrans14=0000000000000000
almRCM_EventTrans15=0000000000000000
almRCMB_EventTrans1=0000000000000000
almRCMB_EventTrans2=0000000000000000
...
almRCMB_EventTrans14=0000000000000000
almRCMB_EventTrans15=0000000000000000
almRMSC_EventTrans1=0000000000000000
almRMSC_EventTrans2=0000000000000000
...
almRMSC_EventTrans14=0000000000000000
almRMSC_EventTrans15=0000000000000000
almSTATION_EventTrans1=0000000000000000
almSTATION_EventTrans2=0000000000000000
...
almSTATION_EventTrans14=0000000000000000
almSTATION_EventTrans15=0000000000000000
almDAANG_EventTrans1=0000000000000000
almDAANG_EventTrans2=0000000000000000
...
almDAANG_EventTrans14=0000000000000000
almDAANG_EventTrans15=0000000000000000
BMS_EventTrans1=0000000000000000
BMS_EventTrans2=0000000000000000
...
BMS_EventTrans14=0000000000000000
BMS_EventTrans15=0000000000000000
```

<sup>3</sup> Note that BaileyOPCServer.INI file also has other sections (e.g. [FC151] section for defining text strings for FC151)

## 8. OPC Alarm and Event Server Interface

The alarm and Event server provides the ability to post alarms and messages to an Alarm summary application type. Alarm and Event server uses a custom interface and supports OPC Alarm and Event Interface Version 1.02.

### 8.1 Event Types

The OPC Server produces the following event types at the A&E interface:

1. **Condition Events** (i.e. Alarms) are sourced from the data tags configured in the OPC server. Alarms are determined within the DCS (i.e. not within the OPC Server) and are “echoed” to the OPC A&E interface
2. **Simple Event** (i.e. Messages) are on a more general information nature, such as communication errors, software faults.
3. **Tracking Events** are not supported.

### 8.2 Common Event Header Information

The following table defines the fields common to each A&E event (i.e. alarm).

Common Header Fields			
Header Field	Description		
Event Type	Condition Event		
Time	Date and time received, taken from clock on hosting PC, in UTC format		
Source	Tag Name from tag database		
Category	<table border="0"> <tr> <td style="vertical-align: top;"> <ul style="list-style-type: none"> <li>▪ Supported category types are:</li> <li>▪ ANALOG - for FC30</li> <li>▪ DIGITAL - for FC45</li> <li>▪ DAANG - for FC177</li> <li>▪ DD - for FC123</li> <li>▪ MSDD - for FC129</li> <li>▪ RCM - for FC62</li> <li>▪ RMCB - for FC136</li> <li>▪ TEXTSTR - for FC194</li> </ul> </td> <td style="vertical-align: top; padding-left: 20px;"> <ul style="list-style-type: none"> <li>▪ STATION – for FC 80, 21, 22, 23</li> <li>▪ Harmony AI – for FC222</li> <li>▪ Harmony AO – for FC223</li> <li>▪ Harmony DI – for FC224</li> <li>▪ Harmony DO – for FC225</li> <li>▪ System - System category is used for sending arbitrary information messages to client applications. There are no associated event attributes.</li> </ul> </td> </tr> </table>	<ul style="list-style-type: none"> <li>▪ Supported category types are:</li> <li>▪ ANALOG - for FC30</li> <li>▪ DIGITAL - for FC45</li> <li>▪ DAANG - for FC177</li> <li>▪ DD - for FC123</li> <li>▪ MSDD - for FC129</li> <li>▪ RCM - for FC62</li> <li>▪ RMCB - for FC136</li> <li>▪ TEXTSTR - for FC194</li> </ul>	<ul style="list-style-type: none"> <li>▪ STATION – for FC 80, 21, 22, 23</li> <li>▪ Harmony AI – for FC222</li> <li>▪ Harmony AO – for FC223</li> <li>▪ Harmony DI – for FC224</li> <li>▪ Harmony DO – for FC225</li> <li>▪ System - System category is used for sending arbitrary information messages to client applications. There are no associated event attributes.</li> </ul>
<ul style="list-style-type: none"> <li>▪ Supported category types are:</li> <li>▪ ANALOG - for FC30</li> <li>▪ DIGITAL - for FC45</li> <li>▪ DAANG - for FC177</li> <li>▪ DD - for FC123</li> <li>▪ MSDD - for FC129</li> <li>▪ RCM - for FC62</li> <li>▪ RMCB - for FC136</li> <li>▪ TEXTSTR - for FC194</li> </ul>	<ul style="list-style-type: none"> <li>▪ STATION – for FC 80, 21, 22, 23</li> <li>▪ Harmony AI – for FC222</li> <li>▪ Harmony AO – for FC223</li> <li>▪ Harmony DI – for FC224</li> <li>▪ Harmony DO – for FC225</li> <li>▪ System - System category is used for sending arbitrary information messages to client applications. There are no associated event attributes.</li> </ul>		
Message	Tag description, from tag database.		
Severity	Severity, in the domain (0,1,2,3,4,5,6), where 0 is the least severe, taken from alarm priority setting within OPC server as follows: 0-Info, 1-LoLo, 2-Low, 3-Medium, 4-High 5-HiHi, 6-Critical		
ConditionName	Same as Category Name		
SubConditionName	Refer to Table in section 8.3		
ChangeMask	Status byte		
NewState	Status byte		
Quality	Refer to Q (OPC Item Status and Quality) in Section 1.7		
AckRequired	Always set to TRUE=1 for all events; Always set to FALSE=1 for all messages		
ActiveTime	System time in file time format		
Actor	Account name for logged on user.		

## 8.3 Supported Sub-Conditions for Events

The supported SubConditions are:

Sub-condition	Default Message	Default Definition	Notes
BQ	Bad Quality	Bad Quality	Reported by hardware on communications error
CFN	Alarm	Digital alarm	DIGITAL, DD, MSDD, RCM, RMCB, TEXTSTR, XDI, XDO
HI	High level alarm	High level limit exceeded	DAANG, Station, Analog, XAI, XAO
LO	Low level alarm	Low level limit exceeded	DAANG, Station, Analog, XAI, XAO
DEV_HI	Deviation High	Deviation limit exceeded	DAANG, Station
DEV_LO	Deviation Low	Deviation limit exceeded	DAANG, Station
HI2	High alarm level 2	High level 2 limit exceeded	DAANG, TEXTSTR
LO2	Low alarm level 2	Low level 2 limit exceeded	DAANG
HI3	High alarm level 3	High level 3 limit exceeded	DAANG, TEXTSTR
LO3	Low alarm level 3	Low level 3 limit exceeded	DAANG
RATE_HI	High rate	Rate limit exceeded	DAANG
RATE_LO	Low rate	Rate limit exceeded	DAANG
RTN	Return to normal	Normal condition	all

The supported SubConditions by Category are:

	ANALOG XAI, XAO	DIGITAL XDI, XDO	DAANG	DD	MSDD	RCM	RMCB	TEXTSTR	STATION
BQ	*	*	*	*	*	*	*	*	*
CFN		*		*	*	*	*		
HI_HI									
LO_LO									
HI	*		*					*	*
LO	*		*						*
HI2			*					*	
LO2			*						
HI3			*					*	
LO3			*						
DEV_HI			*						*
DEV_LO			*						*
RATE_HI			*						*
RATE_LO			*						*
RTN	*	*	*	*	*	*	*	*	*

## 8.4 Event Attributes

In addition to the header, each event has a set of attributes associated with it. These attributes include most, and for some tag types (categories) all, of the fields available at the OPC Data Access Interface.

There is also a field called TYPE, defined as follows:

- TYPE = PE if Category is one of (ANALOG, DIGITAL, DAANG, DD, MSDD, RCM, RMCB, STATION, TEXTSTR, XAI, XAO, XDI, XDO)
- ELSE TYPE = SYSTEM

## 8.5 Message Header Information

The following table defines the fields for all Messages generated at the A&E Interface.

Common Header Fields	
Header Field	Description
Event Type	Simple Event
Time	Date and time received, taken from clock on hosting PC, in UTC format
Source	Subsystem which generated this message
Category	Not Used
Message	Text of information message
Severity	Always 0 = Info for messages.
ConditionName	Not used
SubConditionName	Not used
ChangeMask	Status byte
NewState	Status byte
Quality	Refer to Q (OPC Item Status and Quality) in Section 1.7
AckRequired	Always set to FALSE=1 for all messages
ActiveTime	System time in file time format
Actor	Account name for logged on user.

## 8.6 Message Attributes

There are no attributes to a Message.

# Appendix A – Error Codes

Table A1 provides details of error codes which may occur at the *LastError* OPC Item for CIU, Module of Tag objects. Please contact your supplier if you observe any of these error codes.

Table A1 - Error Codes					
Error	Description	Error	Description	Error	Description
0	No error	28	Index already established (by other computer)	100	Undefined message type
1	Waiting for Loop	29	Point type incompatible with command	101	Busy
2	Improper format	30	Watchdog timeout	102	Mode conflict
3	Illegal Command	31	Checksum compare error	103	Illegal data
4	Index already established	32	Destination Node off-line	104	Invalid block number
5	Block already established another point	33	CALLUP Command required	105	Undefined block number
6	Command too long	34	Computer interface error	106	Block not readable
7	Bad reply from Node Interface	35	Computer interface busy	107	Invalid function code
8	Export Used as Import	36	INBTM01 or INNIS01 offline	108	Function code and block number not compatible
9	Repeat RESTART Command	37	Conflict with Monitor Mode	109	Insufficient memory to write block
10	Undefined Index	38	Point Type	110	Module not responding
11	Memory Full	39	Destination Loop Offline		
12	Host Communication Error	40	Destination Node Busy		
13	INBM01 or INNIS01 not responding	41	Destination Loop Busy		
14	Import used as Export	42	Enhanced trend not established		
15	Timeout of Plant Loop Response				
16	Number out of range			128	Waiting for module reply
17	Illegal Key				
18	Need a restart command				
19	Module status used as Import			255	CIU not responding
20	Message Active on Loop				
21	Import or Export used as Module Status				
22	Exception report specifications lost				
23	No message Queued/Dequeued Received				
24	Reply too large				
25	Illegal station mode command				
26	Illegal Module number in command				
27	Timeout between bytes in command				