Batch.Objects

Q.iMPACT Application Interface Module (For ControlLogix)

Implementation Manual

Private and Confidential

Notices

The information and specifications in this document are subject to change without notice and should not be construed as a commitment by Siempelkamp Canada Inc. Siempelkamp Canada Inc. assumes no responsibility for any errors that may be in this document.

Printed in Canada.

⚠ CAUTION

Because of the variety of uses for the software described in this publication, those responsible for the application and use of this software must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes, and standards.

Contents

Tables

Figures

1 Introduction

1.1 Purpose and Scope

This document describes how to install and configure Siempelkamp's Q.iMPACT Application Interface Module (AIM) in an Allen Bradley ControlLogix. It is intended for those who are installing Mettler-Toledo's Q.iMPACT Matroller(s) and integrating them to an existing application.

The AIM is a PLC software program that allows PLC applications to communicate with Mettler-Toledo's Q.iMPACT Matroller hardware. The AIM provides a framework to easily integrate your application to Mettler-Toledo's Q.iMPACT Matrollers. Configuration and programming are still required in both the AIM and your application software to complete the total solution.

It is assumed that those using this solution have a familiarity with both Mettler-Toledo's Q.iMPACT Matroller and the application software requirements.

1.2 Definitions and Terms

The following terms are used throughout this document:

Table 1: Definitions and Terms

Term	Definition
AIM	Siempelkamp's Q.iMPACT Application Interface Module
Ethernet Terminal Network	The information network that the Q.iMPACT Matrollers are connected to for Web configuration and collaborative data collection.
Cluster	The collective Q.iMPACT Matrollers connected to the Ethernet Terminal Network
ControlNet Network	The fieldbus network that the Q.iMPACT Matrollers are connected to for control purposes.
Bridge	A Q.iMPACT Matroller that is connected as a node on the ControlNet network. A maximum of 12 bridges may exist per Cluster.
Assembly Slot	Each bridge has a maximum of 24 Assembly Slots. Each Assembly Slot connects to 1 Channel.
Channel	A Channel can be configured as either a Scale or a Flow Meter. One such field instrument connects with one Channel.
Q.iMPACT Matroller	A Q.iMPACT Matroller is a Q.iMPACT Material Transfer Controller. Q.iMPACT Matrollers are embedded with the PAC algorithm.
PAC algorithm	A Predictive Adaptive Control algorithm used for precise cut off of Material Transfers.
User Application/EM	The user application or equipment module (EM) software is the PLC application code that is expected to interact with the AIM.
AIM_MAIN	Format for PLC Program Files
Cmd_Data_Assm_Slot	Format for PLC data table variable

1.3 Related Documentation

The following documents act as reference material to this document and should be read and understood prior to implementing an Interceptor solution.

Table 2: Related Documentation

Manufacturer	Description
Mettler Toledo	Q.iMPACT Users Manual and related documentation
Rockwell Automation	ControlNet PLC Programmable Controller documentation

1.4 Compatibility

The following table contains a listing of all hardware tested and found compatible with the AIM software. Please visit the Batch.Objects website at http://www.batchobjects.com for the latest revisions to this table and for additional release notes.

Table 3: Compatibility

Released Version	Date	ControlLogix Firmware Version(s) Supported
V 1.0.0.0	10/20/2002	V 10.0
V 1.1.0.0	06/03/2003	V 10.0
V 1.1.5.0	08/19/2003	V 10.0, V 12.0
V 1.1.6.0	05/20/2004	V 10.0, V 12.0
V 1.1.7.0	06/28/2004	V 10.0, V 12.0
V 1.1.8.0	07/29/2004	V 10.0, V 12.0
V 1.2.0.0	12/07/2004	V 10.0, V 12.0, V 13.0
V 1.2.1.0	02/28/2005	V 10.0, V 12.0, V 13.0

1.5 Memory Requirements

The AIM requires the following memory space.

Table 4: Memory Requirements

Configuration	Memory Allowance
1 Bridge	160 Kb
Additional Bridges	8.4 Kb / Bridge

1.6 Configuration Limits

The AIM has the following configuration limits.

Table 5: Configuration Limits

Item	Limits
Bridges	10
Assembly Slots/Channels per Bridge	24
MPI	1000
Total Number of Instruments	200

2 Overview

The AIM is a set of ControlLogix program files that manage communications to/from one or more Q.iMPACT Matrollers. This section reviews the compatible hardware architectures. It illustrates the software architecture and summarizes the program files and data tables.

2.1 Hardware Architecture

The AIM is flexible, and can handle many hardware configurations. The software may reside in a single PLC and be configured for multiple Bridges, or it may reside in multiple PLCs configured for a single bridge, or any combination between.

This is illustrated in the following figures.

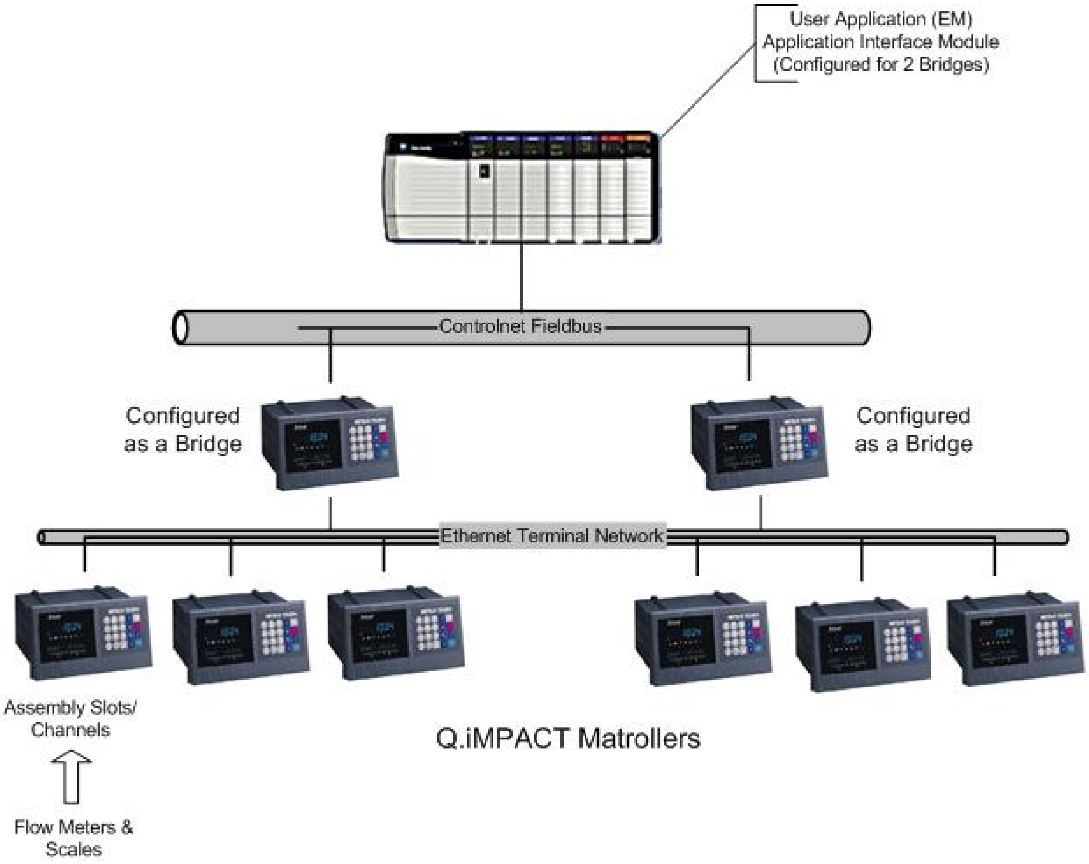


Figure 1: Single Instance AIM Application

The AIM licence allows you to install and configure as many instances of the software as required for a single Q.iMPACT cluster.

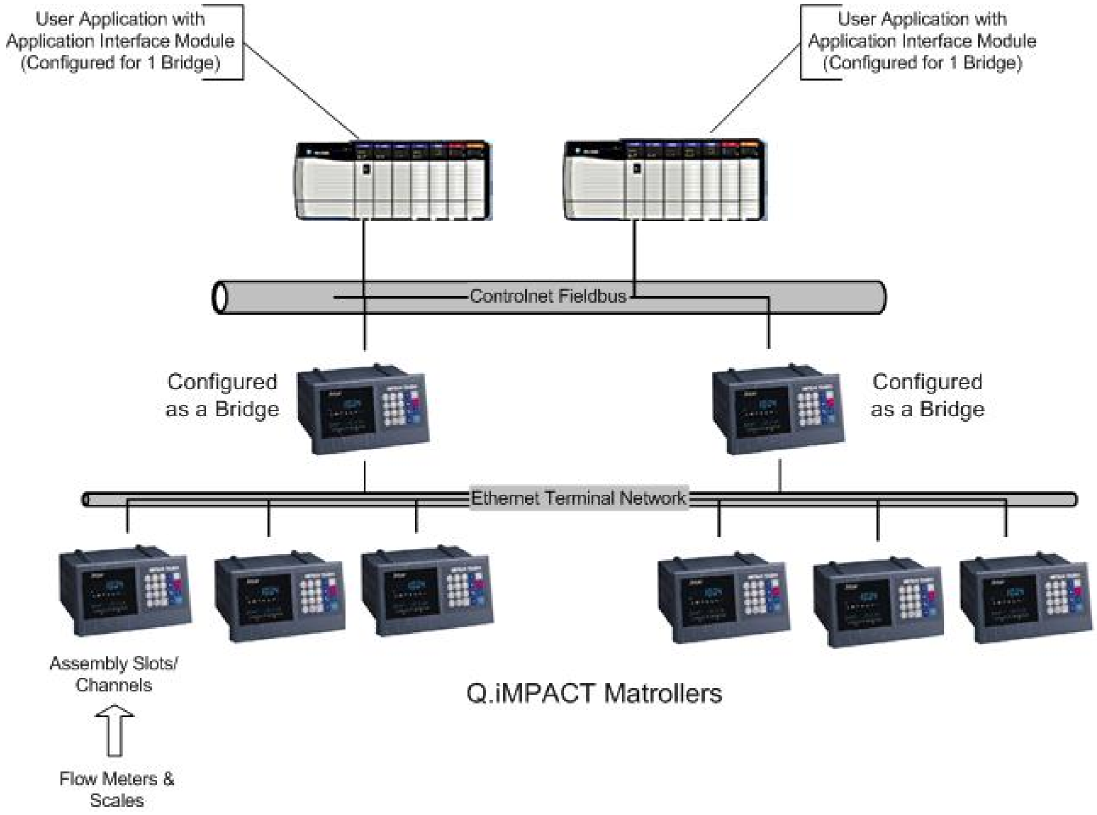


Figure 2: Multiple Instance AIM Application

2.2 Software Architecture

The AIM consists of 11 program files and associated tag data, which must be integrated into your application logic.

The AIM data is organized into three main data structures:

1. AIM Data Structure
2. Bridge Data Structure
3. Instrument Data Structure

These three main data structures are scalable to suit the user application. User application logic interacts with the AIM through Instrument Data. The Bridge data structure manages the transfer of data between the Instrument Data and the communications buffers.

Three communications buffers exist within the AIM:

1. Command Write Buffer
2. Command Read Buffer
3. Cyclic Buffer

These store data transferred between the PLC and the Q.iMPACT Matroller via ControlNet. The following figure illustrates the AIM software architecture.

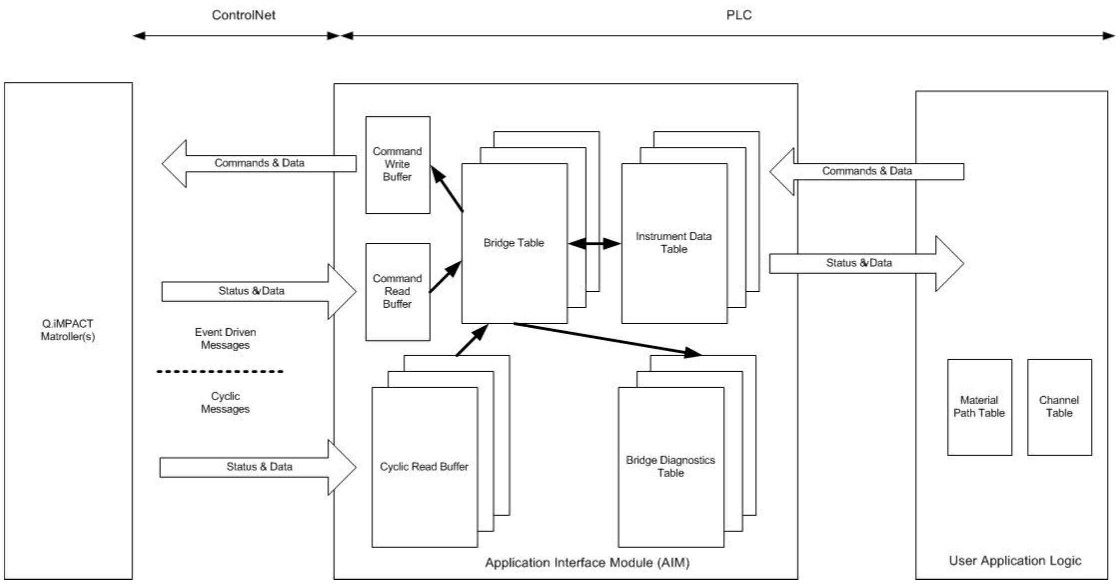


Figure 3: AIM Software Architecture

2.3 Program Files

The AIM consists of 11 program files, which are listed below:

Table 6: Table of Program Files

File Name	File Description
AIMMain	Calls to Main Sub-routines and misc. logic
AIMCommandMain	General Command Logic
AIMCommandProcess	Command Processing Logic
AIMResponse	General Command Response Logic
AIMResponseProcess	Command Response Processing Logic
AIMCyclicMain	General Cyclic Data Logic
AIMCyclicProcess	Cyclic Data Processing Logic
AIMCyclicRead	Cyclic Data Read Logic from Qi *
AIMCyclicCheckSum	Performs Check Sum operation on Cyclic Data
AIMEMInterface	Logic to Monitor Instrument Data Table for New Commands
AIMMSGWrite	MSG Write Instructions *
AIMMSGRead	MSG Read Instructions *

* Unprotected Files that are available to the user.

2.4 Data Table Files

The AIM data is organized into three main data structures:

1. AIM Data Structure
2. Bridge Data Structure
3. Instrument Data Structure

Table 7: The AIM Data Structures

Structure	Description	Data Type
AIM	Application interface module tag structure. Single instance of AIM UDT.	AIM UDT
AIMBridge[#Bridges]	Bridge tag data structure. Multiple instances of Bridge UDT, the number of instances is determined by the required number of Q.i. Bridges in the system architecture (1-10).	Bridge UDT
AIMInstrument[#Instruments]	Instrument tag data structure. Multiple instances of the Instrument UDT, the number of instances is determined by the required number of instruments with the Q.I. Cluster (1-200).	Instrument UDT

There are 20 other control tag addresses reserved for AIM messaging, however, they do not require user interaction.

Table 8: Message Instruction Control Tag Structure

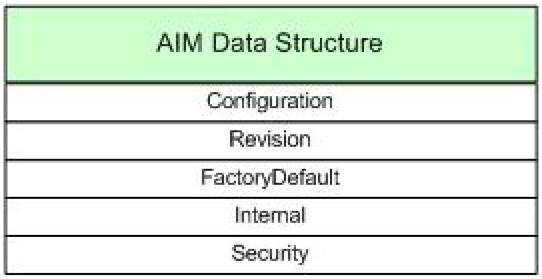
Tags	Description	Data Type
AIMBridge01CnetMessageWrite	Control Net Write Control Registers	Message
.		
.		
.		
AIMBridge10CnetMessageWrite		
AIMBridge01CnetMessageRead	Control Net Read Control Registers	Message
.		
.		
.		
AIMBridge10CnetMessageRead		

2.4.1 AIM Data Structure

The AIM data structure contains all application level data for the Application Interface Module.

This data includes:

- Application level configuration elements
- Revision control information
- Factory default settings
- Internal Registers



AIM Data Structure

Configuration

Revision

FactoryDefault

Internal

Security

Figure 4: AIM Data Structure

2.4.2 Bridge Data Structure

The Bridge data structure contains all bridge level data for the Application Interface Module.

This data includes:

- Communications buffers
- Communications diagnostic data
- Bridge level configuration elements

The Bridge tags exist as instances of the “Bridge” User Defined Data Type (UDT).

The Application Interface Module allows up to 11 instances or arrays of Bridge tags to exist. Array element 0 is used for internal buffering and may not be addressed by the user application. The user application interfaces with array elements 1-10 as configured.

The number of Bridge array elements is determined by the user’s application requirements.

The Bridge array is sized according to the following formula:

(Required # of Bridges + 1)

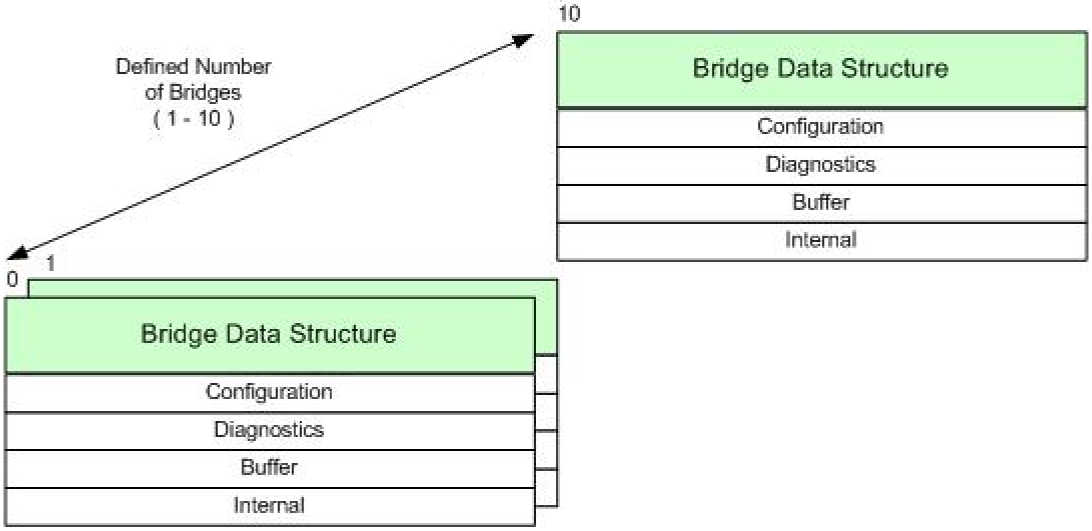


Figure 5: Bridge Data Structure

2.4.3 Instrument Data Structure

The Instrument data structure contains instrument level data for ALL of the Bridges, in the Application Interface Module.

This data includes:

- User application handshake interface
- Command data
- Cyclic data and instrument level configuration elements

The Instrument tag exists as an array of the Instrument User Defined Data Type (UDT).

The Application Interface Module allows up to 201 instances or arrays of Instrument tags to exist. Array element 0 is used for internal buffering and may not be addressed by the user application. The user application interfaces with array elements 1-200 as configured.

The number of Instrument array elements is determined by the user's application requirements. The array is sized according to the following formula:

(Total required # of Instruments on all Bridges + 1)

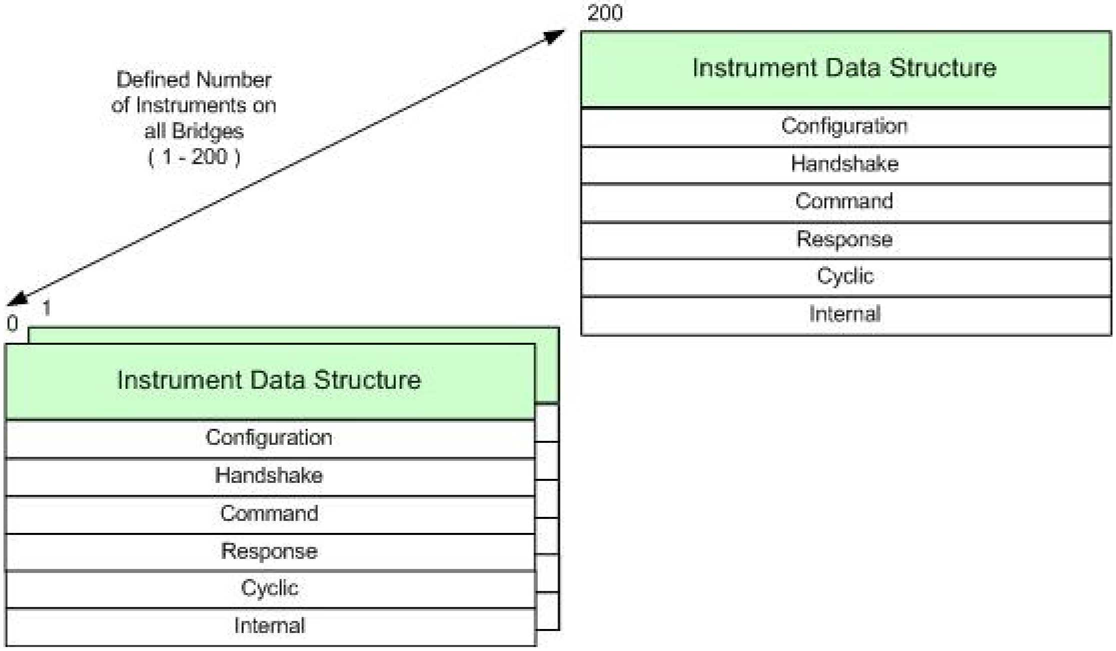


Figure 6: Instrument Data Structure

3 Data Structure Details

The following section describes the details of the three AIM data structures to facilitate integrating your application logic to the AIM.

The three main data structures within the AIM are:

1. AIM Data Structure
2. Bridge Data Structure
3. Instrument Data Structure

The Instrument Data Structure is the primary data structure that the user application logic interacts with. The other data structures are only used at project setup for configuration, or for troubleshooting.

3.1 AIM Data Structure

The AIM data structure contains cluster level data that is used internally by the AIM application. It also contains some application level configuration items available to the user.

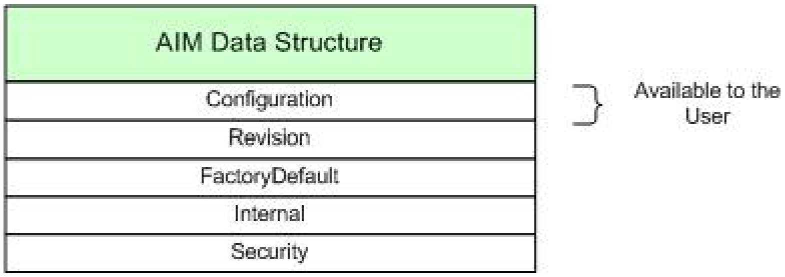


Figure 7: AIM Data Structure

Note: All timer time bases are 0.01 seconds (100= 1 sec)

Other than specifying parameters during configuration, the AIM data should not be used by logic external to the AIM.

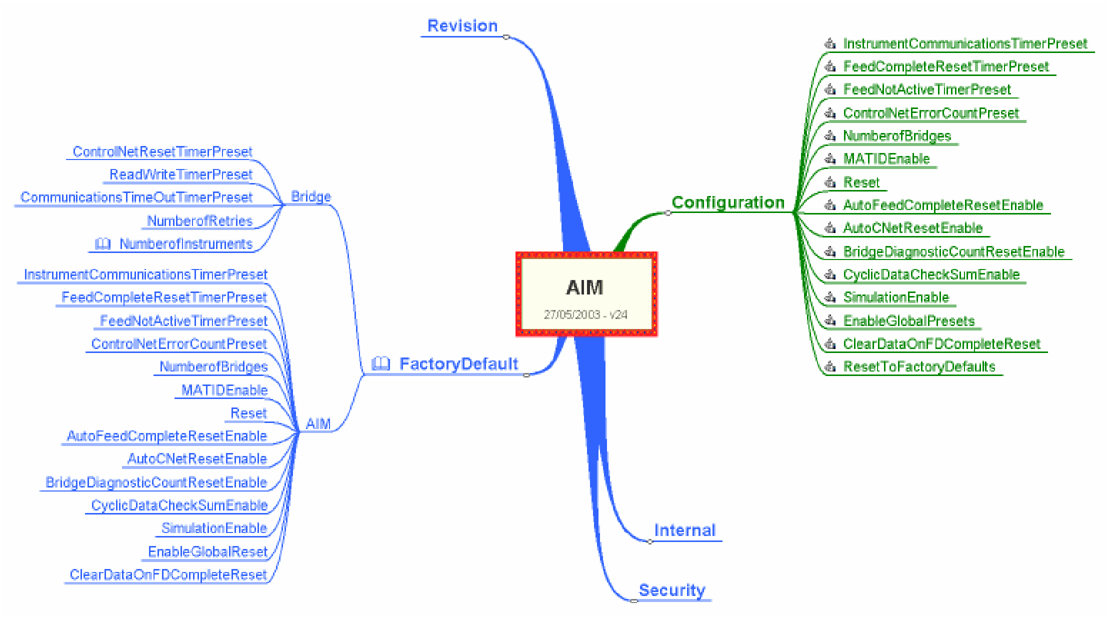


Figure 8: AIM Data Structure

3.1.1 AIM.Configuration Data

This structure contains AIM configuration data.

The AIM.Configuration data is available to the user. See Chapter 5 for details on using the configuration data.

The AIM.Configuration data is outlined in the following figure:

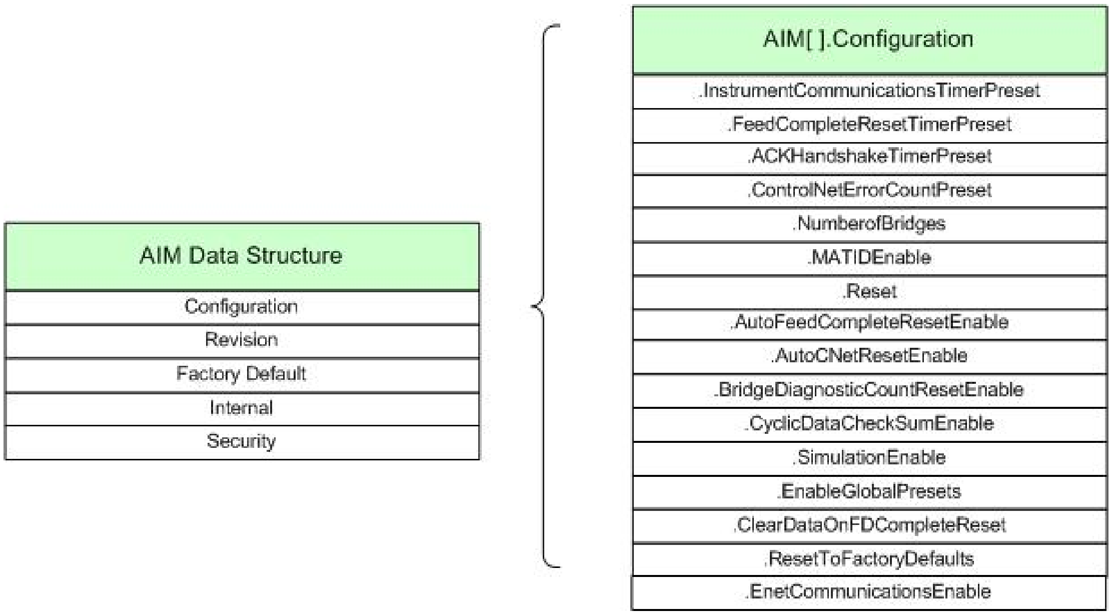

AIM Data Structure
Configuration
Revision
Factory Default
Internal
Security
AIM[ ].Configuration
.InstrumentCommunicationsTimerPreset
.FeedCompleteResetTimerPreset
.ACKHandshakeTimerPreset
.ControlNetErrorCountPreset
.NumberofBridges
.MATIDEnable
.Reset
.AutoFeedCompleteResetEnable
.AutoCNetResetEnable
.BridgeDiagnosticCountResetEnable
.CyclicDataCheckSumEnable
.SimulationEnable
.EnableGlobalPresets
.ClearDataOnFDCompleteReset
.ResetToFactoryDefaults
.EnetCommunicationsEnable

Figure 9: AIM.Configuration Data Structure

Table 9: Data Structure Members

Tag	.Members	Description	Data Type
AIM.Configuration			UDT
	.InstrumentCommunicationsTimerPreset	This defines the period of time between oscillations of the AIMInstrument[ ].Cyclic.Status.ComIntegrity bit from the Q.i.	DINT
	.FeedCompleteResetTimerPreset	This defines the period of time from the setting of the AIMInstrument[ ].Hanshake.FDComplete until it is automatically reset by the AIM. Works in conjunction with AIM.Configuration.AutoFeedCompleteResetEnable	DINT
	.FeedNotActiveDebounceTimerPreset	When enabled, the AIM will detect that a Feed completed un-naturally. This will result in a corresponding error and an AIMInstrument[ ].Handshake.FDComplete. This time preset defines the length of time from the AIMInstrument[ ].Status.CycleActive going low until the AIM completes the feed.	DINT
	.AckHandshakeTimerPreset	This Variable determines the length of time the AIM will wait before re-issuing an Acknowledge command.	DINT
	.CNetResetErrorCountPreset	Preset number of Cnet Errors allowed prior to an automatic reset of the Cnet.	Integer
	.NumberofBridges	Defined number of Bridges to be process by the AIM.	Integer
	.MATIDEnable	Enable MaterialID processing.	Boolean
	.AutoFeedCompleteResetEnable	Enable the automatic (un-conditional) reset of the AIMInstrument[ ].Handshake.FDComplete flag, based on time.	Boolean
	.AutoCNetResetEnable	Enable the automatic reset of the Cnet network. Works in conjunction with AIM.Configuration.ControlNetErrorCountPreset	Boolean
	.BridgeDiagnosticCountResetEnable	Enable the reset of Bridge Diagnostic counts. If enabled these counts are zeroed each hour. When disabled the counts accumulate indefinitely.	Boolean
	.CyclicDataCheckSumEnable	Enable Cyclic Data check sum. This ensures Cyclic Data integrity, and may also be used to ensure proper ControlNet operation.	Boolean
	.SimulationEnable	Enable simulation of Q.I. Interaction (FUTURE).	Boolean
	.Reset	Reset all AIM data, not including Configuration Data. This flag is cleared by the AIM when the Reset is complete.	Boolean
	.ResetToFactoryDefaults	Restore Factory Default configuration settings. This flag is cleared by the AIM when the Reset is complete.	Boolean
	.ClearDataOnFDCompleteReset	When Enabled, the AIM will clear Instrument Data for the Instrument when upon a negative transition of the Instruments AIMInstrument[ ].Handshake.FDComplete flag.	Boolean
	.EnableGlobalPresets	Within the Instrument data structure there are three configurable timer presets. These presets may be individually set for each timer when the AIM.Configuration.EnableGlobalPresets is set, or each instrument can be set to the same value which comes from the AIM.Configuration structure if the	Boolean

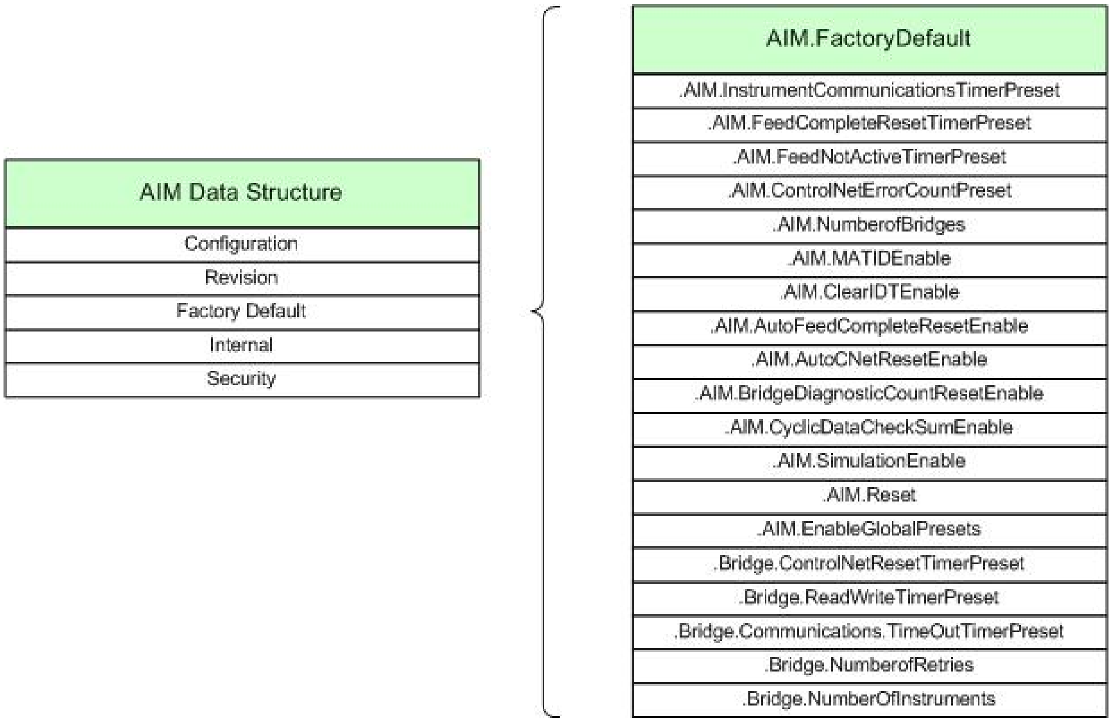
Tag	.Members	Description	Data Type
		AIM.Configuration.EnableGlobalPresets is low.	
	.EnetCommunicationsEnable	This tag enables the AIM to communicate to the Q.i.(s) via Ethernet. 0=ControlNet, 1=Ethernet	Boolean

3.1.2 AIM.FactoryDefault Data

This structure contains factory default settings that can be reset through the configuration registers. See Chapter 5 for configuration details.

The AIM.FactoryDefault data is not available to the user.

The following figure outlines the AIM.FactoryDefault data.



AIM Data Structure

- Configuration
- Revision
- Factory Default
- Internal
- Security

AIM.FactoryDefault

- .AIM.InstrumentCommunicationsTimerPreset
- .AIM.FeedCompleteResetTimerPreset
- .AIM.FeedNotActiveTimerPreset
- .AIM.ControlNetErrorCountPreset
- .AIM.NumberofBridges
- .AIM.MATIDEnable
- .AIM.ClearIDTEnable
- .AIM.AutoFeedCompleteResetEnable
- .AIM.AutoCNetResetEnable
- .AIM.BridgeDiagnosticCountResetEnable
- .AIM.CyclicDataCheckSumEnable
- .AIM.SimulationEnable
- .AIM.Reset
- .AIM.EnableGlobalPresets
- .Bridge.ControlNetResetTimerPreset
- .Bridge.ReadWriteTimerPreset
- .Bridge.Communications.TimeOutTimerPreset
- .Bridge.NumberofRetries
- .Bridge.NumberOfInstruments

Figure 10: AIM.FactoryDefault Data Structure

Table 10: AIM Data Structure Members

Tag	.Members	Description	Data Type
AIM.FactoryDefault			UDT
.AIM			
	.InstrumentCommunicationsTimerPreset	This defines the period of time between oscillations of the AIMInstrument[ ].Cyclic.Status.ComIntegrity bit from the Q.i. .	Integer
	.FeedCompleteResetTimerPreset	This defines the period of time from the setting of the AIMInstrument[ ].Hanshake.FDComplete until it is automatically reset by the AIM. Works in conjunction with AIM.Configuration.AutoFeedCompleteResetEnable	Integer
	.FeedNotActiveTimerPreset	When enabled, the AIM will detect that a Feed completed un-naturally. This will result in a corresponding error and an AIMInstrument[ ].Handshake.FDComplete. This time preset defines the length of time from the AIMInstrument[ ].Status.CycleActive going low until the AIM completes the feed.	Integer
	.ControlNetErrorCountPreset	Preset number of Cnet Errors allowed prior to an automatic reset of the Cnet.	Integer
	.NumberofBridges	Defined number of Bridges to be process by the AIM.	Integer
	.MATIDEnable	Enable MaterialID processing.	Boolean
	.ClearIDTEnable	Enable automatic clearing of Instrument data upon negative transition of the AIMInstrument[ ].Handshake.FDComplete flag.	Boolean
	.AutoFeedCompleteResetEnable	Enable the automatic (un-conditional) reset of the AIMInstrument[ ].Handshake.FDComplete flag, based on time.	Boolean
	.AutoCNetResetEnable	Enable the automatic reset of the Cnet network. Works in conjunction with AIM.Configuration.ControlNetErrorCountPreset	Boolean
	.BridgeDiagnosticCountResetEnable	Enable the reset of Bridge Diagnostic counts. If enabled these counts are zeroed each hour. When disabled the counts accumulate indefinitely.	Boolean
	.CyclicDataCheckSumEnable	Enable Cyclic Data check sum. This ensures Cyclic Data integrity, and may also be used to ensure proper ControlNet operation.	Boolean
	.SimulationEnable	Enable simulation of Q.I. Interaction (FUTURE).	Boolean
	.Reset	Reset all AIM data, not including Configuration Data. This flag is cleared by the AIM when the Reset is complete.	Boolean
	.EnableGlobalPresets	Within the Instrument data structure there are three configurable timer presets. These presets may be individually set for each timer when the AIM.Configuration.EnableGlobalPresets is set, or each instrument can be set to the same value which comes from the AIM.Configuration structure if the AIM.Configuration.EnableGlobalPresets is low.	Boolean
.Bridge			
	.ControlNetResetTimer Preset	The configured length of time to wait, from the point, which a Cnet reset is issued until communications are resumed.	Integer

Tag	.Members	Description	Data Type
	.ReadWriteTimerPreset	The preset length of time between a Cnet MSG write and a Cnet MSG read.	Integer
	.Communications.TimeOutTimerPreset	The maximum length of time between a Cnet MSG Write being issued and a successful Cnet MSG read. If this timer times out the AIMBridge[ ].Internal.RetryCount is incremented.	Integer
	.NumberofRetries	The configured number of MSG Write retries, prior to producing a communications error.	Integer
	.NumberOfInstruments	The defined number of instruments for a given bridge, this is dictated by the user application and the configuration of the Q.iMPACT Matrollers.	Integer

3.2 Bridge Data Structure

The Bridge data structure contains cluster level data that is used internally by the AIM application. It also contains some Bridge level configuration items available to the user.

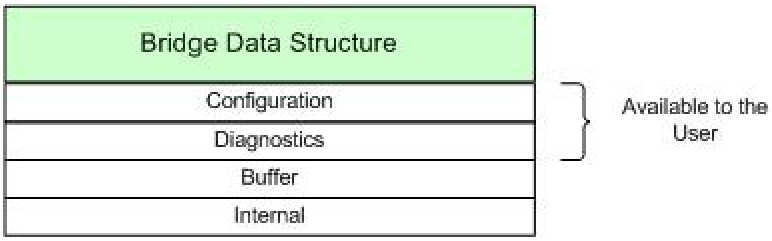


Figure 11: Bridge Data Structure

The Bridge data structure provides the data interface between the Instrument Data Structure and the Q.iMPACT Matroller. The Bridge logic processes data within the AIM to be passed to and from the Q.iMPACT Matroller.

The Q.iMPACT Matrollers support messaging to only one Instrument (Assembly Slot/Channel) per Bridge at a time. Therefore, the Bridge data structure has been designed to process communications to a single Instrument per Bridge at one time. Bridge tags are indexed by Bridge with provisions for up to 10 Bridges as configured.

The array word[0] is not used.

Note: All timer time bases are 0.01 seconds (100= 1 sec)

Other than specifying parameters during configuration, the AIM data should not be used by logic external to the AIM.

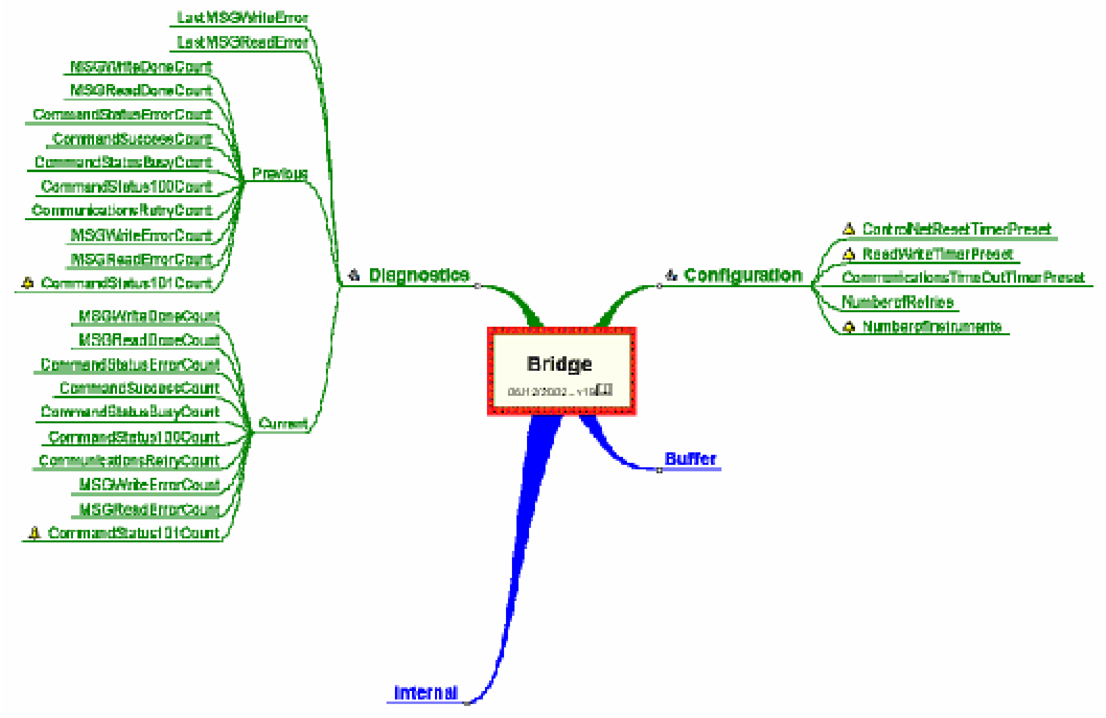


Figure 12: Bridge Data Structure

3.2.1 AIMBridge[ ].Configuration Data

This structure contains Bridge configuration data.

The AIMBridge[ ].Configuration data is available to the user. See Chapter 5 for details on using the configuration data. The AIMBridge[ ].Configuration data is outlined in the following figure:

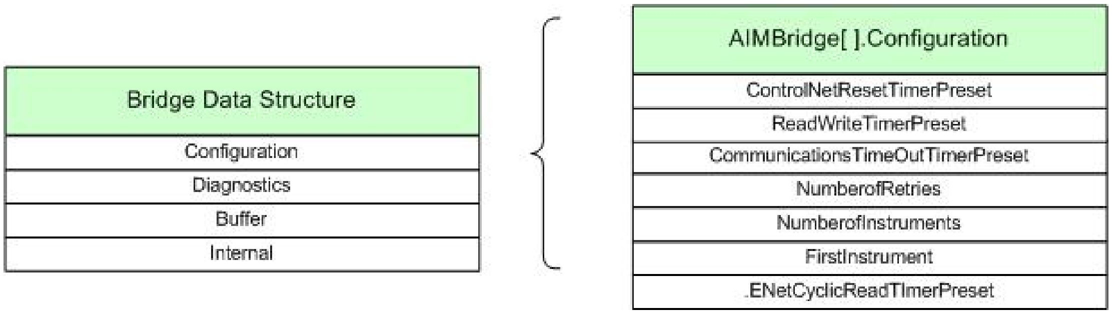


Figure 13: Bridge Configuration Data

The array word [0] is not used within the Bridge data structure. The other array words correspond to the number of bridges configured.

For example, the location of the Read/Write timer preset for Bridge 5 is:

AIMBridge[5].Configuration.ReadWriteTimerPreset

The AIMBridge[ ].Configuration data is listed in the following table.

Table 11: Bridge Data Structure Members

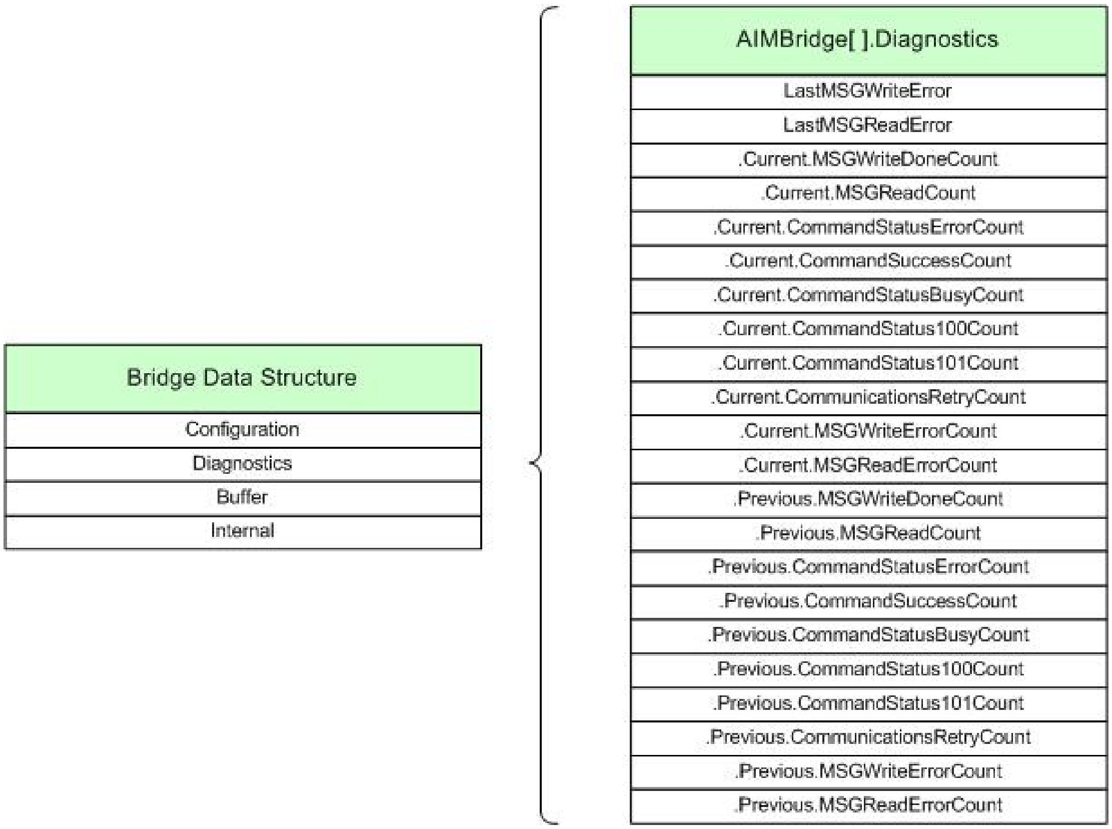
Tag	.Members	Description	Data Type
AIMBridge[ ].Configuration			
	.ControlNetResetTimerPreset	When the AIM issues a ControlNet reset command, ControlNet communications is unavailable while the communications hardware resets. This time is the duration the AIM waits, after a ControlNet reset is issued, before attempting to send commands to the Q.iMPACT Matroller.	DINT
	.WriteReadTimerPreset	The length of time that the AIM waits after sending an explicit write command before issuing the corresponding explicit read command.	DINT
	.CommunicationTimeoutTimerPreset	The length of time the AIM waits, after issuing an explicit write command, for the correct (matching) command response data. If this time is exceeded a communications retry is issued, and AIMBridge[ ].Internal.Handshake.MessageTimeOut is set.	Integer
	.NumberOfRetries	The preset number of attempts, to communicate a command from the AIM to the Q.iMPACT Matroller prior to returning a AIMBridge[ ].Internal.Handhake.MessageResponseMiscompare error, and an AIMInstrument[ ].Response.CommandStatus of 100.	Integer
	.NumberOfInstruments	The defined number of instruments for a given bridge, this is dictated by the user application and the configuration of the Q.iMPACT Matrollers.	Integer
	.ENetCyclicReadTimerPreset	When Ethernet communications are selected, this preset defines the rate at which the cyclic data is updated. Cyclic data is updated via a controller message instruction when Enet is selected.	DINT

3.2.2 AIMBridge[ ].Diagnostic Data

This structure contains diagnostics information that may be useful when troubleshooting.

The AIMBridge[ ].Diagnostic data is available to the user.

The AIMBridge[ ].Diagnostic data structure is shown in the following figure:



Bridge Data Structure

Configuration

Diagnostics

Buffer

Internal

AIMBridge[ ].Diagnostics

LastMSGWriteError

LastMSGReadError

.Current.MSGWriteDoneCount

.Current.MSGReadCount

.Current.CommandStatusErrorCount

.Current.CommandSuccessCount

.Current.CommandStatusBusyCount

.Current.CommandStatus100Count

.Current.CommandStatus101Count

.Current.CommunicationsRetryCount

.Current.MSGWriteErrorCount

.Current.MSGReadErrorCount

.Previous.MSGWriteDoneCount

.Previous.MSGReadCount

.Previous.CommandStatusErrorCount

.Previous.CommandSuccessCount

.Previous.CommandStatusBusyCount

.Previous.CommandStatus100Count

.Previous.CommandStatus101Count

.Previous.CommunicationsRetryCount

.Previous.MSGWriteErrorCount

.Previous.MSGReadErrorCount

Figure 14: Bridge Diagnostic Data Structure

The array word [0] is not used within the Bridge data structure. The other array words correspond to the number of bridges configured.

For example, the location of the command status success count, current hour for Bridge 2 is:

AIMBridge[2].Diagnostics.Current.CommandSuccessCount

The diagnostic data within the Bridge data structure is populated by the AIM.

The AIMBridge[ ].Diagnostic data is listed in the following table.

Table 12: Bridge Data Structure Members

Tag	.Members	Description	Data Type
AIMBridge[ ].Diagnostics			UDT
	.LastMsgWriteError	Returns the communication channels error register after each explicit write command.	Integer
	.LastMSGReadError	Returns the communication channels error register after each explicit read command.	Integer
.Current		Contains diagnostic counts for the current hour.	
	.MSGWriteDoneCount	Each time a successful explicit write command is made to the Bridge this counter is incremented.	Integer
	.MSGReadDoneCount	Each time a successful explicit read command is made to the Bridge this counter is incremented.	Integer
	.CommandStatusErrorCount	Each time the Q.iMPACT returns a Command Status error (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommandStatusSuccessCount	Each time the Q.iMPACT returns a Command Status success (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommandStatusBusyCount	Each time the Q.iMPACT returns a Command Status busy (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommandStatus100Count	Each time the AIM returns a Command Status 100 (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommunicationsRetryCount	Communications retry count, current hour.	Integer
	.MSGWriteErrorCount	Each time an unsuccessful explicit write command is made to the Bridge this counter is incremented.	Integer
	.MSGReadErrorCount	Each time an unsuccessful explicit read command is made to the Bridge this counter is incremented	Integer
	.CommandStatus101Count	Each time the AIM returns a Command Status 101 (See Command Status Classifications within this document) this count is incremented.	Integer
.Previous		Contains diagnostic counts for the previous hour.	
	.MSGWriteDoneCount	Each time a successful explicit write command is made to the Bridge this counter is incremented.	Integer
	.MSGReadDoneCount	Each time a successful explicit read command is made to the Bridge this counter is incremented.	Integer
	.CommandStatusErrorCount	Each time the Q.iMPACT returns a Command Status error (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommandStatusSuccessCount	Each time the Q.iMPACT returns a Command Status success (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommandStatusBusyCount	Each time the Q.iMPACT returns a Command Status busy (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommandStatus100Count	Each time the AIM returns a Command Status 100 (See Command Status Classifications within this document) this count is incremented.	Integer
	.CommunicationsRetryCount	Communications retry count, current hour.	Integer
	.MSGWriteErrorCount	Each time an unsuccessful explicit write command is made to the Bridge this counter is incremented.	Integer
	.MSGReadErrorCount	Each time an unsuccessful explicit read command is made to the Bridge this counter is incremented	Integer
	.CommandStatus101Count	Each time the AIM returns a Command Status 101 (See Command Status Classifications within this document) this count is incremented.	Integer

3.3 Instrument Data Structure

The Instrument Data Structure contains instrument/channel level data and is the interface between the AIM and the user application logic.

Instrument tags are indexed from 1 up to 200 with an assignment for bridge/Instrument combination. There are provisions for 10 bridges and 24 Instruments per bridge within the AIM up to the limit of 200 Instruments.

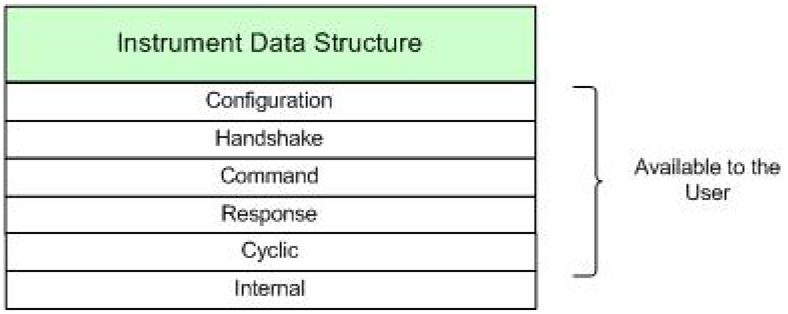


Figure 15: Instrument Data Structure

The array word[0] is not used.

Note: All timer time bases are 0.01 seconds (100= 1 second)

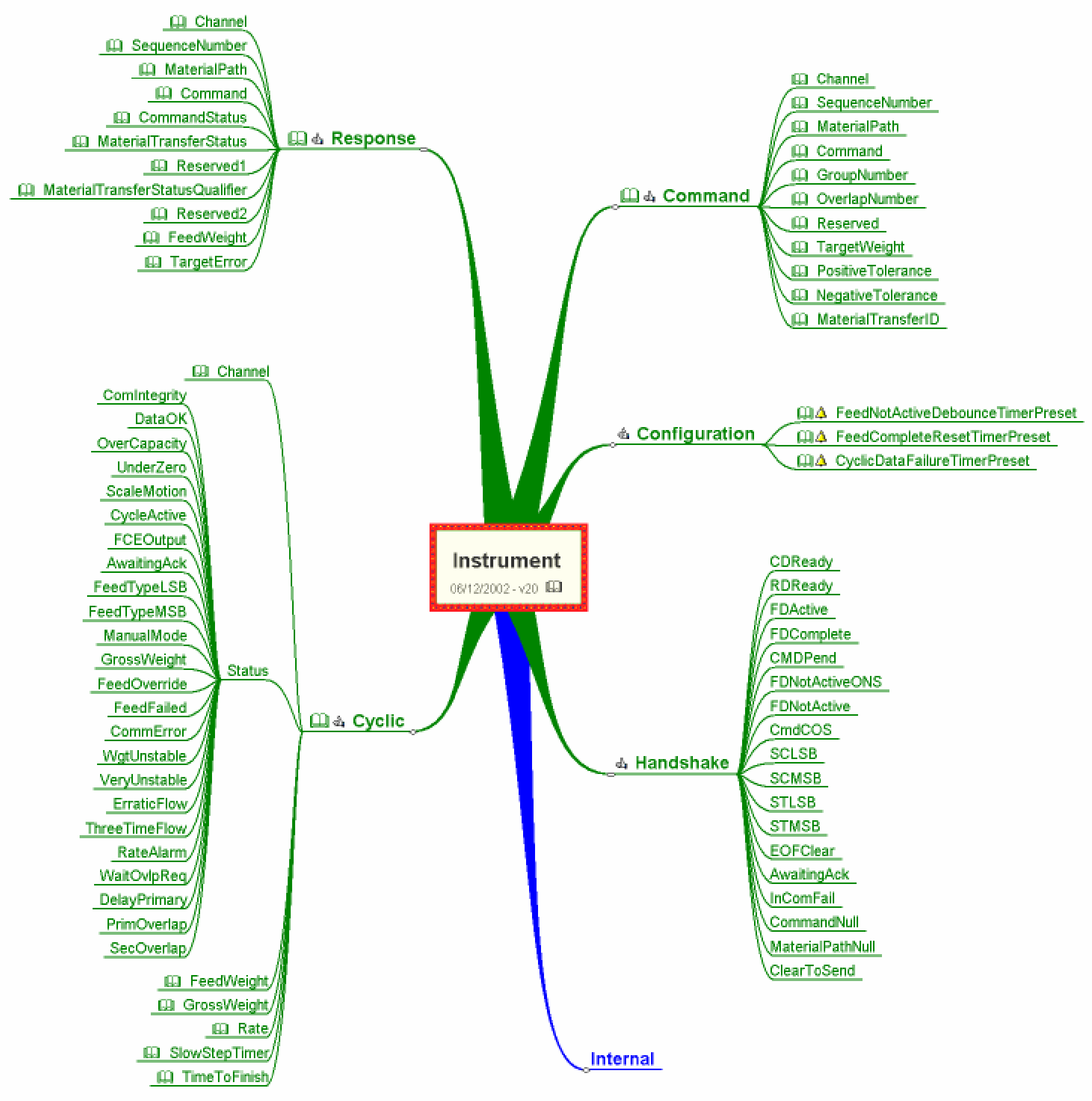


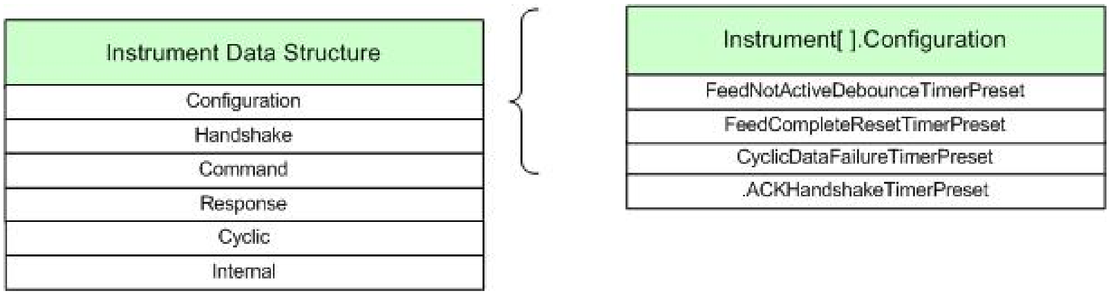
Figure 16: Instrument Data Structure

3.3.1 AIMInstrument[ ].Configuration

This structure contains Instrument configuration data.

The AIMInstrument[ ].Configuration data is available to the user.

The following figure outlines the AIMInstrument[ ].Configuration data.



Instrument Data Structure
Configuration
Handshake
Command
Response
Cyclic
Internal

Instrument[ ].Configuration
FeedNotActiveDebounceTimerPreset
FeedCompleteResetTimerPreset
CyclicDataFailureTimerPreset
.ACKHandshakeTimerPreset

Figure 17: Instrument Configuration Data Structure

The following table summarizes how Instrument configuration data is organized within the AIM.

Table 13: AIMInstrument[ ].Configuration Data Structure Members

Tag	.Members	Description	Data Type
AIMInstrument[ ].Configuration			
	.FeedNotActiveDebounceTimerPreset	In order to eliminate unpredictable operation due to minor communications issues, the AIM debounces the status of the AIMInstrument[ ].Cyclic.CycleActive flag for a period defined by the Feed Not Active Debounce Timer Preset. This control is used to determine whether a feed is active or not, and therefore the true status must be known.	DINT
	.FeedCompleteResetTimerPreset	If the feed complete flag (AIMInstrument[ ].Handshake.FDComplete) remains on for a period longer than defined by the Feed Complete Reset Timer Preset then the AIM automatically resets the flag. Normally the user application would reset this flag.	DINT
	.CyclicDataFailureTimerPreset	The AIM monitors the oscillation of the AIMInstrument[ ].Cyclic.Status.ComIntegrity flag, which is sent to the AIM via cyclic data. If the flag does not oscillate within the period defined by the Cyclic Data Failure Timer Preset, then the AIMInstrument[ ].Handshake.InComFail flag is set to indicate that there is a problem communicating to a certain Instrument.	DINT
	.ACKHandshakeTimerPreset	This Variable determines the length of time the AIM will wait before re-issuing an Acknowledge command.	DINT

3.3.2 AIMInstrument[ ].Handshake Data

This structure contains the Q.iMPACT instrument handshaking data and is a key table for interaction to the user's application.

The AIMInstrument[ ].Handshake data is available to the user.

The following figure outlines the AIMInstrument[ ].Handshake data.

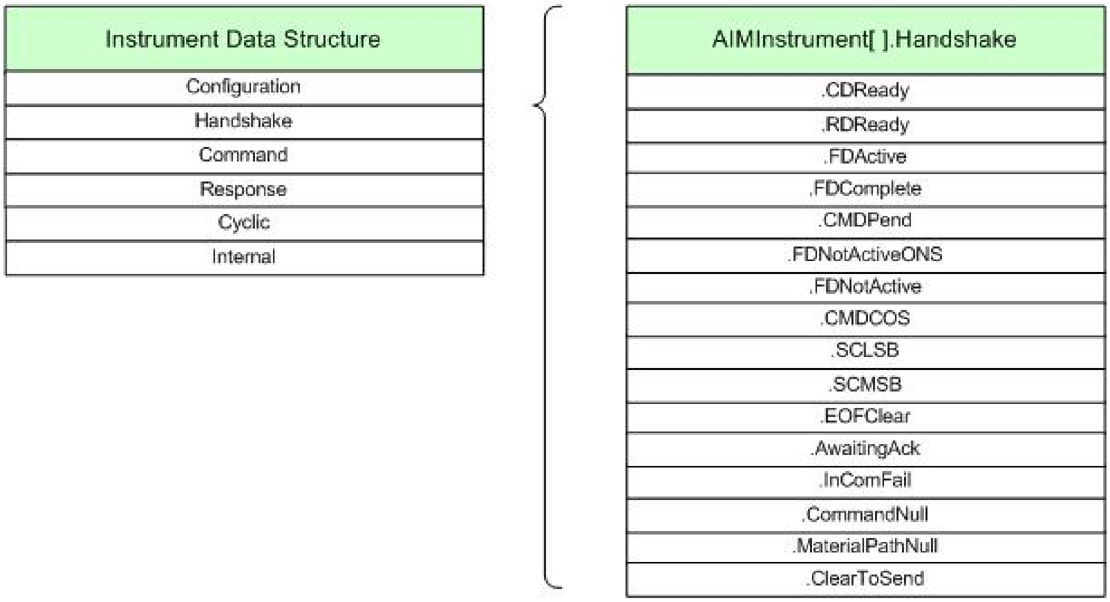


Figure 18: Instrument Handshake Data Structure

The AIMInstrument[ ].Handshake tag members are summarized in the following table.

Table 14: AIMInstrument[ ].Handshake Data Structure Members

Tag	.Members	Description	Set	Reset
AIMInstrument[ ].Handshake				
	.CDReady	Command Data Ready. User Application must ensure this bit is low before transferring data to the instrument table. Once CDReady is low, the AIM must write command data to the Instrument Data Table. The AIM resets CDReady upon completion of the communications handshake with the Q.iMPACT Matroller. Note: The AIM controls the reset of CDReady; therefore under no circumstances should a User Application reset CDReady.	EM/AIM	AIM
	.RDReady	Response Data Ready. Turned off by the AIM when CDReady is set high.	AIM	AIM
	.FDActive	Feed Active. Retrieved from the Q.iMPACT Matroller cyclic data FDActive.	AIM	AIM
	.FDComplete	Feed Complete. The feed complete flag is set by the AIM once it has acknowledged a feed complete from the Q.iMPACT Matroller . The feed complete flag also is set if a command level error is received from the Q.iMPACT Matroller during a Command Read. The FDComplete flag notifies the User Application that the end of feed data is available in the Instrument Data Table. Once the User Application has processed this data, it must reset the FDComplete flag. Once reset, the AIM zero’s all fields for that instrument within the Instrument Data Table.	AIM	EM
	.CMDPend	Command Pending. Once a command from the Instrument Data Table has been processed, the Bridge sets CMDPend in order internally interlock the AIM to prevent another command from being issued to the same Bridge. During this period all processing of commands is suspended, any command issued to the AIM will be queued.	AIM	AIM
	.FDNotActiveONS	Feed Not Active One Shot. Used to capture FDActive going negative.	AIM	AIM
	.FDNotActive	Feed Not Active Interlock. On from time Feed Active goes low unit FDComplete goes high.	AIM	AIM
	.CmdCOS	Communications Change of State	AIM	AIM
	.SCLsb	Status Class LSB These handshake flags are multiplexed to derive the command Status Class. 1 – Success, 2 – Error, 3 – Warning.	AIM	AIM
	.SCMsb	Status Class MSB These handshake flags are multiplexed to derive the command Status Class. 1 – Success, 2 – Error, 3 – Warning.	AIM	AIM

Tag	.Members	Description	Set	Reset
	.STLsb	Status Type LSB These handshake flags are multiplexed to derive the command Status Type. 1 – Bad Data, 2 – Process Error, 3 – Instrument Failure.	AIM	AIM
	.STMsb	Status Type MSB These handshake flags are multiplexed to derive the command Status Type. 1 – Bad Data, 2 – Process Error, 3 – Instrument Failure.	AIM	AIM
	.EOFClear	End of Feed Clear Data. ONS which when initiated by the AIM zero's all the instrument data within the AIMInstrument[ ].Command and AIMInstrument[ ].Response tag data structure for a given instrument.	AIM	AIM
	.AwaitingAck	Waiting for Acknowledge request	AIM	AIM
	.InComFail	Instrument Communications Failure. When the cyclic data communication integrity bit for a given instrument fails, the Instrument Communications Failure flag is set. The Application Logic should condition all commands and shared data requests with this flag to ensure commands are not issued to an instrument that is not communicating.	AIM	AIM
	.CommandNull	Command Zero, This bit is set when a command of 0 is sent to the AIM and is reset by CDReady bit.	AIM	AIM
	.MaterialPathNull	MaterialPath Zero, This bit is set when a MaterialPath of 0 is sent and reset by CDReady bit.	AIM	AIM
	.ClearToSend	When all Instrument interlocks are in a state where a command may be sent to that Instrument, the ClearToSend flag will be set. The User application should use this as an interlock for setting the AIMInstrument[ ].Hanshake.CDReady flag.	AIM	AIM
	.GWILIW	Set when a Material Transfer is active, indicates whether the Material Transfer is a Gain In Weight or a Loss In Weight	AIM	AIM

Note: Several of these bits are mirrored within the AIMBridge[ ].Internal.Handshake data structure.

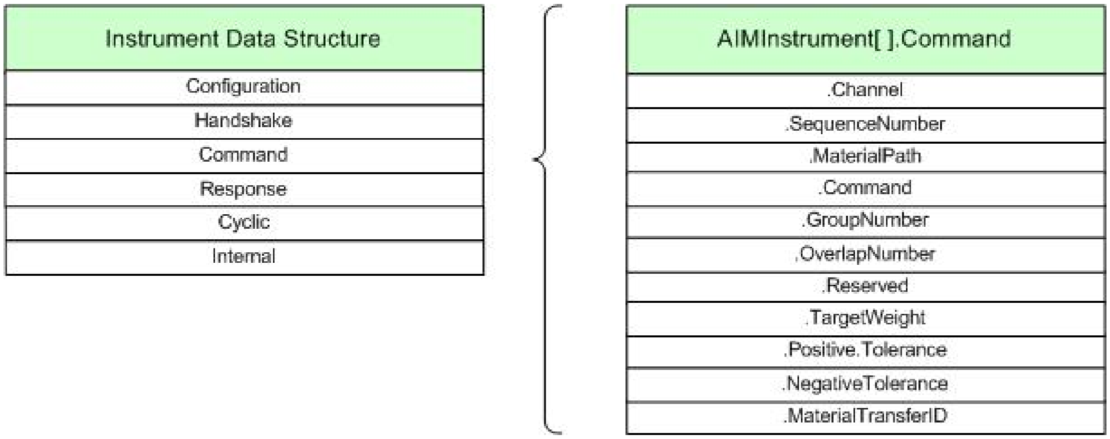
Note: A single handshake data structure exists for each instrument (Assembly Slot/Channel) within the Q.iMPACT Matroller(s).

3.3.3 AIMInstrument[ ].Command Data

This structure contains Q.iMPACT Command data and is a key table for interaction to the user’s application.

The AIMInstrument[ ].Command data is available to the user.

The following figure outlines the AIMInstrument[ ].Command data.



Instrument Data Structure
Configuration
Handshake
Command
Response
Cyclic
Internal

AIMInstrument[ ].Command
.Channel
.SequenceNumber
.MaterialPath
.Command
.GroupNumber
.OverlapNumber
.Reserved
.TargetWeight
.Positive.Tolerance
.NegativeTolerance
.MaterialTransferID

Figure 19: Instrument Command Data Structure

The AIMInstrument[ ].Command tag members are summarized in the following table.

Table 15: AIMInstrument[ ].Command Data Structure Members

Tag	.Members	Description	Data Type
AIMInstrument[ ].Command			
	.Channel	The Channel reflects the Channel assigned to the instrument, which occupies a certain assembly slot within the Q.iMPACT Matroller configuration. The channel for a particular instrument is received within Cyclic data and updated automatically, and therefore does not need to be modified.	
	.SequenceNumber	Contains the AIM generated sequence number, which is the unique identifier sent to the Q.iMPACT Matroller. Each time a command is sent on a particular Bridge.	
	.MaterialPath	MaterialPath Material Path Index (MPI) (1- 1000), reflects the Material Path(s) setup for a certain instrument within the Q.iMPACT Matroller configuration. Cleared by the AIM after the equipment module has acknowledged a transfer complete or a reset.	
	.Command	Command number, which dictates the action to be performed by the Q.iMPACT Matroller when the command (explicit write) data is sent.	
	.GroupNumber	“Group Number” for Start Material Transfer Command. This field identifies which Primary and Secondary feed requests belong to the group of feeds that make up an overlapping feed. A value =? 0 indicates that this NOT an overlapped feed request. Cleared by the AIM when the Material Path is cleared	
	.OverlapNumber	“Number of Overlapping Secondary Feeds” that are being fed into a Unit simultaneously with THIS Primary Feed. This field is only meaningful in a Primary Feed using scale instrument for the Vessel. The Material-Path must indicate this Gain-In-Weight feed. The APC turns on the FCE for the Primary Feed when it determines there will be enough time after the overlap completes to run the APC algorithm with the scale. If this is a zero then this is a feed that is part of the overlap and not the primary feed. Cleared by the AIM when the Material Path is cleared.	
	.Reserved	Reserved	
	.TargetWeight	“Target Weight” For most commands, this field contains the target weight for the feed. For “Turn On FCE In Manual Mode”, the “Target Weight” Field must contain the number of seconds to keep the FCE on” A zero for FCE indicates infinite time.	
	.PositiveTolerance	PosTolerance “Tolerance +”	
	.NegativeTolerance	NegTolerance “Tolerance –“	
	.MaterialTransferID	“Material Transfer ID” is an identifier field that is sent from the Material Transfer Controller. The Q.iMPACT reports this field as part of the data collection record. If there is a “~” in the field, the Q.iMPACT displays the data following the “~”.	

3.3.3.1 AIMInstrument[ ].Command.Command Data

This tag contains the command signals that may be sent to a particular instrument via the AIM. In order to issue a command, the corresponding command value must be written by the user application to this tag in the Instrument Tag Data Structure.

For example, the master instrument reset command for the 24th Instrument (Assembly Slot/Channel) is indicated by:

.AIMInstrument[24].Command = 12

The following table summarizes the commands supported by the AIM and the Q.iMPACT Matrollers.

Table 16: AIMInstrument[ ].Command.Command Word Signals

Value	Command	Name
0	None	None
1	Start Material Transfer	MTStart
2	Start Material Transfer with Gross Weight Target. This command is only valid for a scale device.	GWStart
3	Start Hand Add	HAStart
4	Acknowledge Material Transfer or Hand Add Complete	MTAck
5	Abort Material Transfer	MTAbort
6	Reset Slow Step Timer	SSTReset
7	Start Manual Mode	ModeManual
8	Turn on FCE in Manual Mode	FCEOn
9	Turn off FCE in Manual Mode	FCEOff
10	Restart Auto Mode	ModeAuto
11	Complete Feed in Manual Mode	CFManual
12	Master Reset – Instrument	InstrumentReset
13	Report Last Status	StatusReport
14	Master Reset – Cluster	ClusterReset
15	Validate Aggregate Secondary Feeds	SFValidate
30	ControlNet Reset	CnetReset

Note: The AIM processes the requested command and passes that command to the Q.iMPACT Matroller. A good knowledge of the Q.iMPACT Matroller commands and their operation is required.

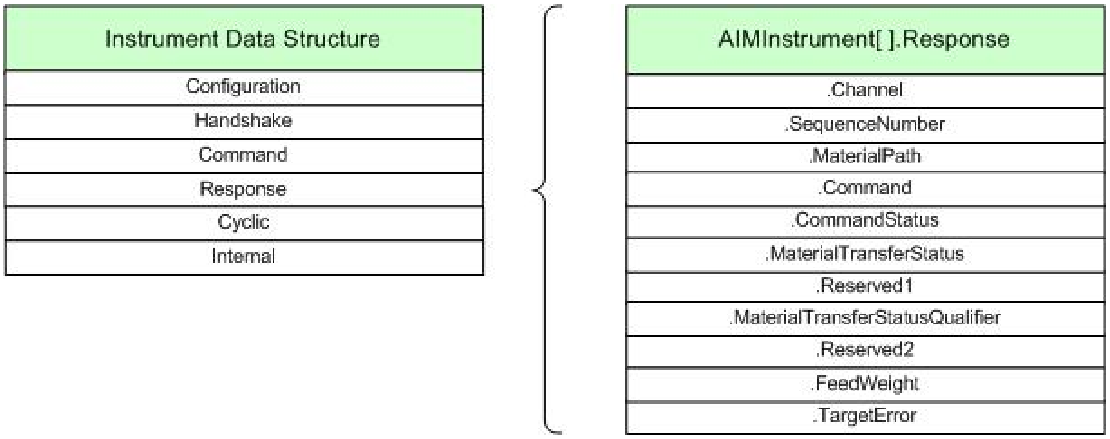
The AIM does not check for invalid commands - *with the exception of a command value of zero*. All other commands will be processed and passed to the Q.iMPACT Matroller. The Q.iMPACT Matroller itself will provide a returned command status of 9, if it receives an invalid command.

3.3.4 AIMInstrument[ ].Response Data

This structure contains Q.iMPACT Response data and is a key table for interaction to the user’s application.

The AIMInstrument[ ].Response data is available to the user.

The following figure outlines the AIMInstrument[ ].Response data.



Instrument Data Structure
Configuration
Handshake
Command
Response
Cyclic
Internal

AIMInstrument[ ].Response
.Channel
.SequenceNumber
.MaterialPath
.Command
.CommandStatus
.MaterialTransferStatus
.Reserved1
.MaterialTransferStatusQualifier
.Reserved2
.FeedWeight
.TargetError

Figure 20: Instrument Response Data Structure

The AIMInstrument[ ].Response tag members are summarized in the following table.

Table 17: AIMInstrument[ ].Response Data Structure Members

Tag	.Members	Description	Data Type
AIMInstrument[ ].Response			
	.Channel	The Channel reflects the Channel assigned to the instrument, which occupies a certain assembly slot within the Q.iMPACT Matroller configuration. The channel for a particular instrument is received within Cyclic data and updated automatically, and therefore does not need to be modified.	
	.SequenceNumber	Contains the AIM generated sequence number, which is the unique identifier sent to the Q.iMPACT Matroller. Each time a command is sent on a particular Bridge.	
	.MaterialPath	Material Path Index (MPI) (1- 1000), reflects the Material Path(s) setup for a certain instrument within the Q.iMPACT Matroller configuration. Cleared by the AIM after the equipment module has acknowledged a transfer complete or a reset.	
	.Command	Command number which dictates the action to be performed by the Q.iMPACT Matroller when the command (explicit write) data is sent.	
	.CommandStatus	The Command Status returned (explicit read) in a command response, reflects the status of the Q.iMPACT Matroller when it receives the command.	
	MaterialTransferStatus	Returned only when an Acknowledge is sent the Q.iMPACT Matroller (the acknowledge is automatically generated by the AIM), the Material Transfer Status reflects the status of the Material Transfer.	
	.Reserved1	Data allocated within the AIM and Q.iMPACT, which is reserved for future use.	
	.MaterialTransferStatusQualifier	Returned only when an Acknowledge is sent the Q.iMPACT Matroller (the acknowledge is automatically generated by the AIM), the Material Transfer Status Qualifier is either 0=Over Weight or 1=Under Weight, with reference to the last Material Transfer.	
	.Reserved2	Data allocated within the AIM and Q.iMPACT, which is reserved for future use.	
	.FeedWeight	Returned only when an Acknowledge is sent the Q.iMPACT Matroller (the acknowledge is automatically generated by the AIM), Feed Weight indicates the amount fed during the last Material Transfer.	
	.TargetError	Returned only when an Acknowledge is sent the Q.iMPACT Matroller (the acknowledge is automatically generated by the AIM), the Target Error reflects the deviation between the Target Weight from the command, and the Feed Weight.	

3.3.4.1 AIMInstrument[ ].Response.MaterialTransferStatus Data

This tag contains material transfer status data from the Q.iMPACT Matroller. It updates once a material transfer is completed, and is returned with the end of feed data.

The following table summarizes the possible material transfer statuses received from the Q.iMPACT Matroller.

Table 18: AIMInstrument[ ].Response.MaterialTransferStatus Data

Value	Status	Description
0	Success	Successful Material Transfer – K1, K2 Parameters Updated
1	Success	Successful Material Transfer – Spill Only
2	Success	Successful Material Transfer – Dump to Empty
3	Success	Successful Material Transfer – Hand Add
4	Success	Material Transfer Complete – Parameters Not Updated
5	Success	Material Transfer Complete – Parameters Reset
6	Success	Material Transfer Complete with Manual Operation
7	Failed	Unstable Scale
8	Failed	Overlapping Feed Error Corrupted Flow
9	Failed	Erratic Flow Error
10	Failed	Low Flow Error
11	Failed	High Flow Rate Alarm Error
12	Failed	Communications Error
13	Failed	Instrument Error
14	Failed	Scale Device Capacity Error
15	Failed	Predictive Algorithm Error
16	Failed	Material Transfer with Manual Operation
17	Failed	Amount of Material transferred did not match in source ad destination
18	Failed	Controller Aborted Material Transfer
19	Failed	Controller Reset Channel
20	Failed	Controller Reset Cluster
21	Failed	Reserved
22	Failed	Slow Step Timer Timeout
23	Failed	Secondary Requests Timeout
24	Failed	Power Failure During Feed
25	Failed	Start Material Transfer Command Failed Immediately – Transfer Did Not Start.
26	Status Only	Material Transfer Is In Progress

3.3.4.2 AIMInstrument[ ].Response.MaterialTransferStatusQualifier Data

This tag contains material transfer status qualifier data from the Q.iMPACT Matroller, which updates after a material transfer completes.

The following table summarizes the possible material transfer status qualifiers received from the Q.iMPACT Matroller.

Table 19: AIMInstrument[ ].Response.MaterialTransferStatusQualifier Data

Bit	Description
0	Over Tolerance
1	Under Tolerance

Bit	Description
2	Power Failure During Feed
3-15	Reserved

3.3.4.3 AIMInstrument[ ].Response.CommandStatus Data

This tag contains the response signals from the Q.iMPACT Matroller to the command signals sent via the AIM.

For example, a successful command complete status for Instrument (Assembly Slot/Channel) 3 is indicated by:

AIMInstrument[3].Response.CommandStatus = 5

The following table summarizes the possible command responses received from the Q.iMPACT Matroller

Table 20: AIMInstrument[ ].Response.CommandStatus Signals

Value	Status Class	Description	Status Type
0	SUCCESS	Start Gain In Weight Material Transf. Command Complete	Success
1	SUCCESS	Start Loss In Weight Material Transfer Complete.	Success
2	SUCCESS	Start Flow Meter Material Transfer Complete.	Success
3	SUCCESS	Start Manual Control Material Transfer Complete.	Success
4	SUCCESS	Start Hand Add Command Complete.	Success
5	SUCCESS	Command Complete	Success
6		Command Not Complete – Request status again after a short delay	
7	ERROR	Communications Error	Bad Data
8	ERROR	Invalid Instrument Number	Bad Data
9	ERROR	Invalid Command	Bad Data
10	ERROR	Invalid Material-Path Table Index Number	Bad Data
11	ERROR	Invalid Algorithm in Material-Path Table Entry	Bad Data
12	ERROR	Invalid Feed Type in Material-Path Table Entry	Bad Data
13	ERROR	Invalid Measuring Device Instrument Table Index in Material Path Table Entry	Bad Data
14	ERROR	Invalid Gain In Weight Feed and Dump to Empty Algorithm Combination in Material Path Table	Bad Data
15	ERROR	Invalid Destination in Material Path Table Entry.	Bad Data
16	ERROR	Other invalid data in Material Path Table Entry	Bad Data
17	ERROR	Overlap Feed Request Error, including invalid Loss In Weight Feed in Material Path Entry and Overlapping Feed Command.	Bad Data
18	ERROR	Invalid data In Measuring Device Instrument Table Entry	Bad Data
19	ERROR	Invalid Mode for Command, e.g., PLC is requesting to start a new material transfer before the last feed is complete or before the PLC has acknowledged that the last material transfer is complete.	Process Error
20	ERROR	Requested add amount too small	Process Error

Value	Status Class	Description	Status Type
21	ERROR	Requested add amount would bring Scale Device over capacity	Process Error
22	ERROR	Scale Device Currently over Capacity	Process Error
23	ERROR	Scale Device Currently under Zero.	Process Error
24	ERROR	Instrument Malfunction	Instrument
25	ERROR	Target Weight is less than Spill	Process Error
26	ERROR	Response Timeout	Instrument
27	ERROR	Too many overlapping feeds	
28	WARNING	Delayed start to feed due to overlapping feed.	
29	WARNING	Abort ignored since Time to Complete was less than Feed Override Time	
30	ERROR	Invalid overlap group number	Bad Data
31	WARNING	Waiting for All Secondary Requests	
32	WARNING	Waiting for Measuring Device Stability.	
33	ERROR	Not Enough Material	Process Error
34	ERROR	Flow Meter Configuration Error	Bad Data
100	ERROR	Command Failed	Communication
101	ERROR	Material Transfer Ended Without Request for Acknowledge	Communication

Note: Command Status values of 100 and above are generated by the AIM, all other Command status values are generated by the Q.iMPACT Matroller.

3.3.5 AIMInstrument[ ].Cyclic Data

This structure contains Q.iMPACT Cyclic Read data.

The AIMInstrument[ ].Cyclic data is available to the user.

The following figure outlines the AIMInstrument[ ].Cyclic data.

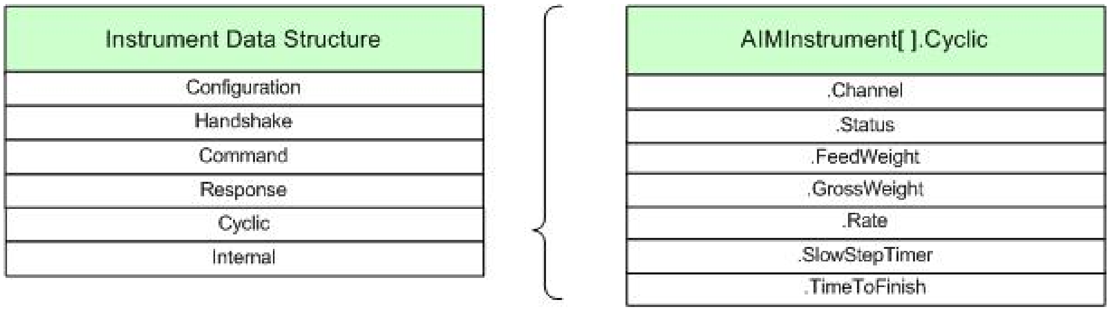


Figure 21: Instrument Cyclic Data Structure

The AIMInstrument[ ].Cyclic tag members are summarized in the following table.

Table 21: AIMInstrument[ ].Cyclic Data Structure Members

Tag	.Members	Description	Data Type
AIMInstrument[ ].Cyclic			
	.Channel	The Channel reflects the Channel assigned to the instrument, which occupies a certain assembly slot within the Q.iMPACT Matroller configuration. "Instrument Number" (1 – 200)	
	.Status	Returned at regular intervals the Status data dynamically indicates various instrument data.	
	.FeedWeight	Returned at regular intervals the Feed Weight dynamically indicates the amount fed during a Material Transfer During most feeds, this field contains the Net Accumulated Weight for the single feed. During Primary Overlapped feeds, this field contains the combined weight of all feeds. At completion of the feed, it contains the Delivered Weight for this feed.	
	.GrossWeight	Returned at regular intervals the Gross Weight dynamically indicates the Gross Weight calculated by the instrument.	
	.Rate	Returned at regular intervals the Rate dynamically indicates the Flow rate calculated by the instrument. "Rate of Change of Weight"	
	.SlowStepTimer	Returned at regular intervals the Slow Step Timer dynamically indicates the value of the Slow Step Timer (watch dog timer) "Time until Slow Step Timer Expires" in Seconds	
	.TimeToFinish	Returned at regular intervals the Time to Finish dynamically indicates the remaining time for a particular feed calculated by the Q.iMPACT Matroller.	

3.3.5.1 AIMInstrument[ ].Cyclic.Status Data

Three bytes of data are returned from the Q.iMPACT Matroller via the Cyclic Read Buffer. This status information contains instrument related indications generated within the Q.iMPACT Matroller.

For example, a feed failed status on Instrument (Assembly Slot/Channel) 7 is indicated by:

AIMInstrument[7].Cyclic.Status.FeedFailed = 1

This data structure is summarized in the following table.

Table 22: AIMInstrument[ ].Cyclic.Status – Data Layout

Name	Description
.ComIntegrity	Communications Data Integrity Bit alternates Polarity Each Second
.DataOK	Instrument Data Integrity OK
.OverCapacity	Scale Over Capacity
.UnderZero	Scale Under Zero
.ScaleMotion	Scale Motion
.CycleActive	Material Transfer Cycle Active
.FCE_Output	Final Control Element Output 0 = Off, 1 = On
.AwaitingACK	Waiting for PLC to Acknowledge Last Material Transfer/Hand Add complete
.FeedTypeLSB	Feed Type
.FeedTypeMSB	0=Gain In Weight, 1= Loss In Weight, 2= Flow Meter, 3 = Hand Add
.ManualMode	Manual-Not Auto-Mode
.GrossWeight	Gross Weight Feed
.FeedOverride	Feed Override Active – external logic inhibited from removing feed permissive
.FeedFailed	Feed Failed
.CommError	Communication Error
.WgtUnstable	Device Stability Warning
.VeryUnstable	Very Unstable Device
.ErraticFlow	Too High or Too Low Flow at cutoff
.3TimesFlow	Three Times Average Flow at cutoff
.RateAlarm	Fast Feed Rate Alarm
.WaitOvlpReq	Wait for All Overlap Requests
.DelayPrimary	Waiting to Start Primary Overlapped Feed
.PrimOverlap	Primary Overlapped Feed In Progress
.SecOverlap	Secondary Overlapped Feed In Progress

4 Processing

This section describes the communications processing that occurs in the AIM.

AIM Communications to and from a Q.iMPACT Matroller uses both scheduled and event based messaging.

4.1 Scheduled Communications

Scheduled communications use the Cyclic Buffer to update the Instrument Data Structure. The User Application does not initiate this, but it does monitor the Instrument Data Structure for status.

Status information is continuously received from each Q.iMPACT Bridge according to the ControlNet update rate that is defined for that scheduled cyclic message, or if Ethernet communications is selected the rate is defined by the Bridge[ ].configuration.

The status data is buffered within the AIM until the data for an entire Instrument is decoded prior to publishing the data to the Instrument Data Structure. This prevents partial data from .ENetCyclicReadTimerPreset being present in the Instrument Data Structure, which may lead to an improper User Application operation.

4.1.1 Communications Integrity

The AIM monitors the AIMInstrument[ ].Cyclic.Status.ComIntegrity status indicator within the Instrument Data Structure. This status indicator oscillates between 1 and 0, at five second intervals. Each time Cyclic Data is processed, the AIM monitors the time that this status indicator has been off. If the status indicator remains off (i.e. Cyclic Data fails to turn on the status indicator) for the preset length of time, the timer times out and the AIMInstrument[ ].Handshake.InComFail status indicator is set in the Instrument Data Structure.

This error is reported in the handshaking word of each Instrument (assembly slot), for the related instrument within the Q.iMPACT Matroller. This error is reset by the AIM when communications with that particular instrument resumes. While this error is present, the AIM does not allow commands for that instrument to be processed.

The communications timer has a default preset value of 15 000 milliseconds.

The location of the Instrument Communications Failure Timer Preset is:

AIMInstrument[ ].Configuration.CyclicDataFailureTimerPreset

The location of the Instrument Data Structure handshake data is:

AIMInstrument[ ].Handshake

4.2 Event Based Communications

Event based communications uses the Command Write Buffer and Command Read buffer within the AIM to update the Instrument Data Structure. The User Application initiates event based communications and must monitor status.

The User Application initiates commands to an instrument via the Instrument Data Structure.

To initiate a command to a Q.iMPACT Matroller, the User's Application will:

1. Write the command data to the appropriate registers within the Instrument Data Structures Command data, AIMInstrument[ ].Command.
2. Set the Command Data Ready (CDReady) flag in the instruments handshake data, AIMInstrument[ ].Handshake.CDReady.

When a response is received, the AIM will:

3. Verify the response with the associated command

Upon verification, the AIM will:

4. Write response data to the Instrument Data Structures Response data, AIMInstrument[ ].Response.
5. Set the Response Data Ready (AIMInstrument[ ].Handshake.RDReady) flag within the instrument handshake data,
6. Reset the command data ready (AIMInstrument[ ].Handshake.CDReady) flag,.

The Users Application can reset the AIMInstrument[ ].Handshake.RDReady control once the response data has been read and processed. However, the AIM will reset the AIMInstrument[ ].Handshake.RDReady control before each new command is sent to the Q.iMPACT Matroller. This ensures that there is a low to high transition of the AIMInstrument[ ].Handshake.RDReady control each time Command Read data is received. If the user application is not designed to handle response handshaking, then this control can be ignored.

The message response data is buffered within the AIM until the entire message is decoded prior to publishing the data to the Instrument Data Structure. This prevents partial data from being present in the Instrument Data Table, which may lead to improper User Application operation.

Commands are issued by the AIM to the Q.iMPACT Matroller via MSG Write commands, and Command Read data is received via a MSG Read commands.

4.2.1 A Typical Feed

The following table and figure illustrate a typical feed sequence.

Table 23: Typical Feed Sequence

Step	Who	Action	Matroller
1	User Application	Checks that the AIMInstrument[ ].Handshake.CDReady flag is not set. Writes command data to the Instrument Data Structure, (AIMInstrument[ ].Command) Sets the AIMInstrument[ ].Handshake.CDReady Flag	Idle
2	AIM	Sets AIMInstrument[ ].Handshake.CMDPend Initiates MSG write command Starts MSG write/read timer once the corresponding bridge is clear	Idle
3	AIM	During the period between the MSG write and the associated MSG read performed by the AIM, the Cyclic Data will return the AIMInstrument[ ].Cyclic.Status.CycleActive , and the AIM will echo the AIMInstrument[ ].Handshake.FeedActive to the instrument handshake data.	Feed Active
4	AIM	Perform MSG Read Command Read data is made available to the User Application in the Instrument Data Structure - such as Active MPI and Command Status. Sets the AIMInstrument[ ].Handshake.RDReady flag Resets the AIMInstrument[ ].Handshake.CDReady flag	Feed Active
5	AIM	When the feed is complete, the Q.iMPACT Matroller will turn off Feed Active (AIMInstrument[ ].Cyclic.Status.CycleActive) and turn on Awaiting Feed Complete Acknowledge (AIMInstrument[ ].Cyclic.Status.AwaitingAck), within the Cyclic Data for the instrument.	Feed Complete
6	AIM	Upon receiving the Awaiting Feed Complete Acknowledge indication from the Q.iMPACT Matroller, the AIM writes Ack Command (command = 4) to AIMInstrument[ ].Command.Command and sets the AIMInstrument[ ].Handshake.CDReady flag within the instrument handshake word.	Feed Complete
7	AIM	Sets AIMInstrument[ ].Handshake.CMDPend Resets AIMInstrument[ ].Handshake.CDReady flag Initiates MSG write command Starts MSG write/read timer once the corresponding bridge is clear.	Feed Complete
8	AIM	MSG Read The Command Read data is made available to the User Application in the Instrument Data Structure. Sets the AIMInstrument[ ].Handshake.RDReady flag Resets the AIMInstrument[ ].Handshake.CDReady flag. AIMInstrument[ ].Handshake.FDComplete flag is set to signify that the feed is complete and end of feed data is available in the Instrument Data Structure, and in the Q.iMPACT Matroller.	Feed Complete
9	User Application	Processes the end of feed data Resets the AIMInstrument[ ].Handshake.FDComplete flag	Idle

Step	Who	Action	Matroller
	AIM	Initializes the End of Feed Clear (AIMInstrument[ ].Handshake.EOFClear) flag.	Idle
10	AIM	Clears all data for the instrument, which has just completed a feed within the Instrument Data Structure (Configurable).	Idle

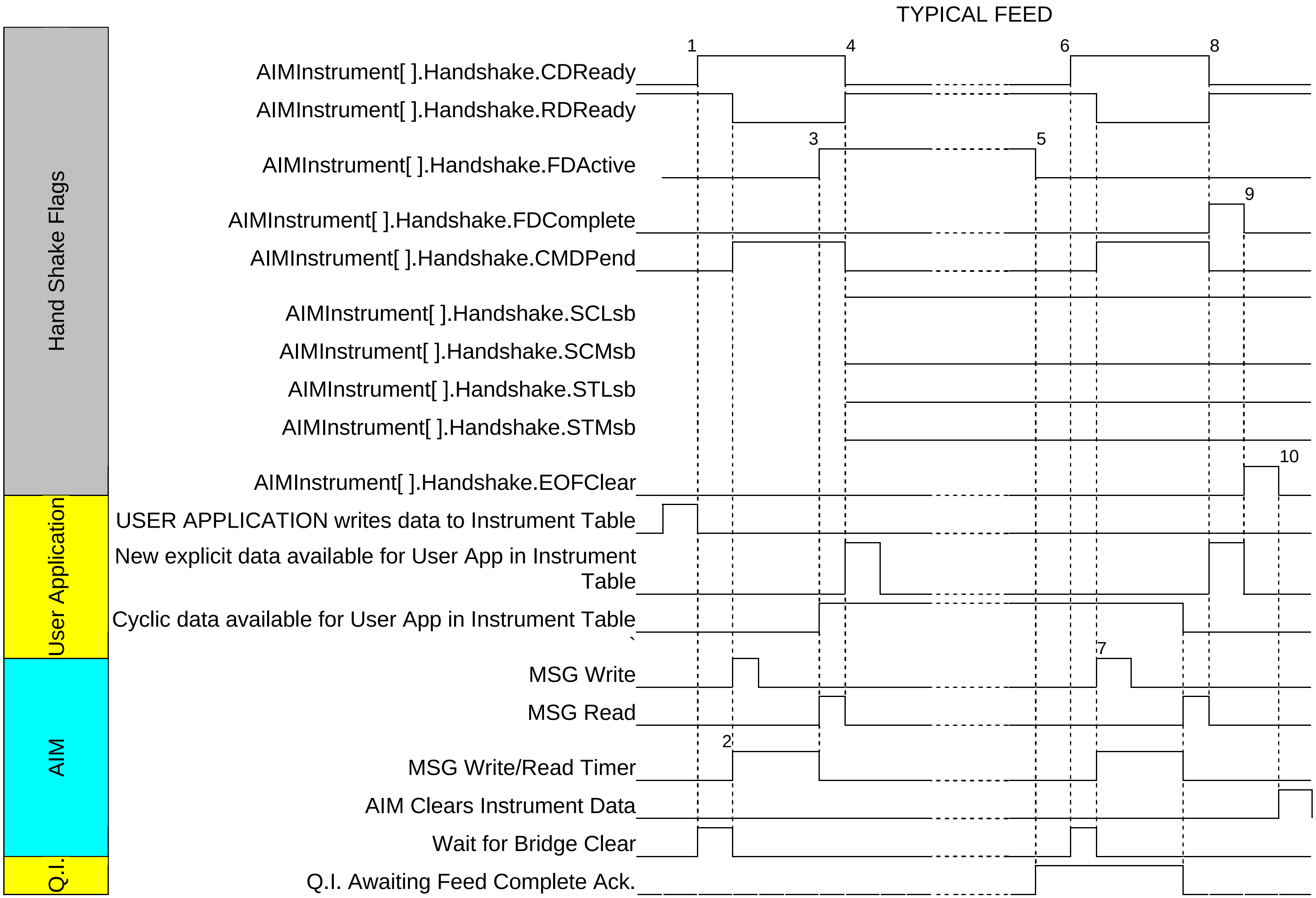

TYPICAL FEED
Hand Shake Flags
User Application
AIM
Q.I.
AIMInstrument[ ].Handshake.CDReady
AIMInstrument[ ].Handshake.RDReady
AIMInstrument[ ].Handshake.FDActive
AIMInstrument[ ].Handshake.FDComplete
AIMInstrument[ ].Handshake.CMDPend
AIMInstrument[ ].Handshake.SCLsb
AIMInstrument[ ].Handshake.SCMsb
AIMInstrument[ ].Handshake.STLsb
AIMInstrument[ ].Handshake.STMsb
AIMInstrument[ ].Handshake.EOFClear
USER APPLICATION writes data to Instrument Table
New explicit data available for User App in Instrument Table
Cyclic data available for User App in Instrument Table
MSG Write
MSG Read
MSG Write/Read Timer
AIM Clears Instrument Data
Wait for Bridge Clear
Q.I. Awaiting Feed Complete Ack.
1
2
3
4
5
6
7
8
9
10

Figure 22: Typical Feed Sequence

(Reference numbers within the figure refer to steps in the previous table.)

4.2.2 A Typical Failure

The following table and figure illustrate a typical failed sequence.

Table 24: Typical Failure

Step	Who	Action	Matroller
1	User Application	Checks that the AIMInstrument[ ].Handshake.CDReady flag is not set Writes command data to the Instrument Data Structure, (AIMInstrument[ ].Command) Sets the AIMInstrument[ ].Handshake.CDReady Flag Note: Commands and subsequent retries are not executed while an InComFail error exists. Once communications is restored, commands and/or retries will resume.	State is Unknown
2	AIM	Sets AIMInstrument[ ].Handshake.CMDPend Initiates MSG write command Starts MSG write/read timer once the corresponding bridge is clear Command Time Out Timer begins timing	State is Unknown
3	AIM	After the MSG Read/Write timer has timed out, the MSG read command is issued.	State is Unknown
4	AIM	In this example, the matching Command Read is not received by the AIM, and the Command Time Out Timer will continue timing until it times out. At this point the AIMBridge[ ].Internal.Handshake.MessageTimeOut flag is set, the command pending flag is reset and the retry count is incremented.	State is Unknown
5	AIM	Since the AIMInstrument[ ].Handshake.CDReady Flag is still high, the AIM re-initiates the previous command. The AIM sets Command Pending, resets the AIMBridge[ ].Internal.Handshake.MessageTimeOut flag and waits for the Bridge to be clear.	State is Unknown
6	AIM	Once the Bridge is clear to send a command, the AIM initiates the MSG write command and starts the MSG write/read timer. In addition the Command Time Out Timer begins timing within the AIM.	State is Unknown
7	AIM	After the MSG Read/Write timer has timed out, the MSG read command is issued.	State is Unknown
8	AIM	If for any reason the matching Command Read is not received by the AIM, the Command Time Out Timer will continue timing until it times out. At this point the AIMBridge[ ].Internal.Handshake.MessageTimeOut flag is set, the command pending flag is reset and the retry count is incremented.	State is Unknown
9	User Application	Since the AIMInstrument[ ].Handshake.CDReady Flag is still high, the AIM re-initiates the previous command. The AIM sets Command Pending, resets the AIMBridge[ ].Internal.Handshake.MessageTimeOut flag and waits for the Bridge to be clear. This pattern (Steps 5 to 8) will be repeated until the retry count equals the user configured retry count for the bridge.	State is Unknown

Step	Who	Action	Matroller
10	AIM	If the retry count equals the configured number of retries, an AIMBridge[ ].Internal.Handshake.MessageResponseMiscompare error is generated by the AIM. At this point an AIMInstrument[ ].Response.CommandStatus of 100, AIMInstrument[ ].Response.TargetError of the command AIMInstrument[ ].Command.TargetWeight and AIMInstrument[ ].Response.MaterialStatusQualifier of 1 (under tolerance) is written to the Instrument Data structure.	State is Unknown
11	AIM	Sets the AIMInstrument[ ].Handshake.RDReady flag Resets the AIMInstrument[ ].Handshake.CDReady flag AIMInstrument[ ].Handshake.FDComplete flag is set to signify that the feed is complete and end of feed data is available in the Instrument Data Table.	State is Unknown
12	User Application	Once the User Application has processed the end of feed data, it resets the AIMInstrument[ ].Handshake.FDComplete flag.	State is Unknown
	AIM	At this point, the AIM initializes the End of Feed Clear (AIMInstrument[ ].Handshake.EOFClear) flag and resets the AIMBridge[ ].Internal.Handshake.MessageResponseMiscompare flag.	State is Unknown
13	AIM	When the AIMInstrument[ ].Handshake.EOFClear flag is set, the AIM clears all data for the instrument which has just completed a feed within the Instrument Data structure.	State is Unknown

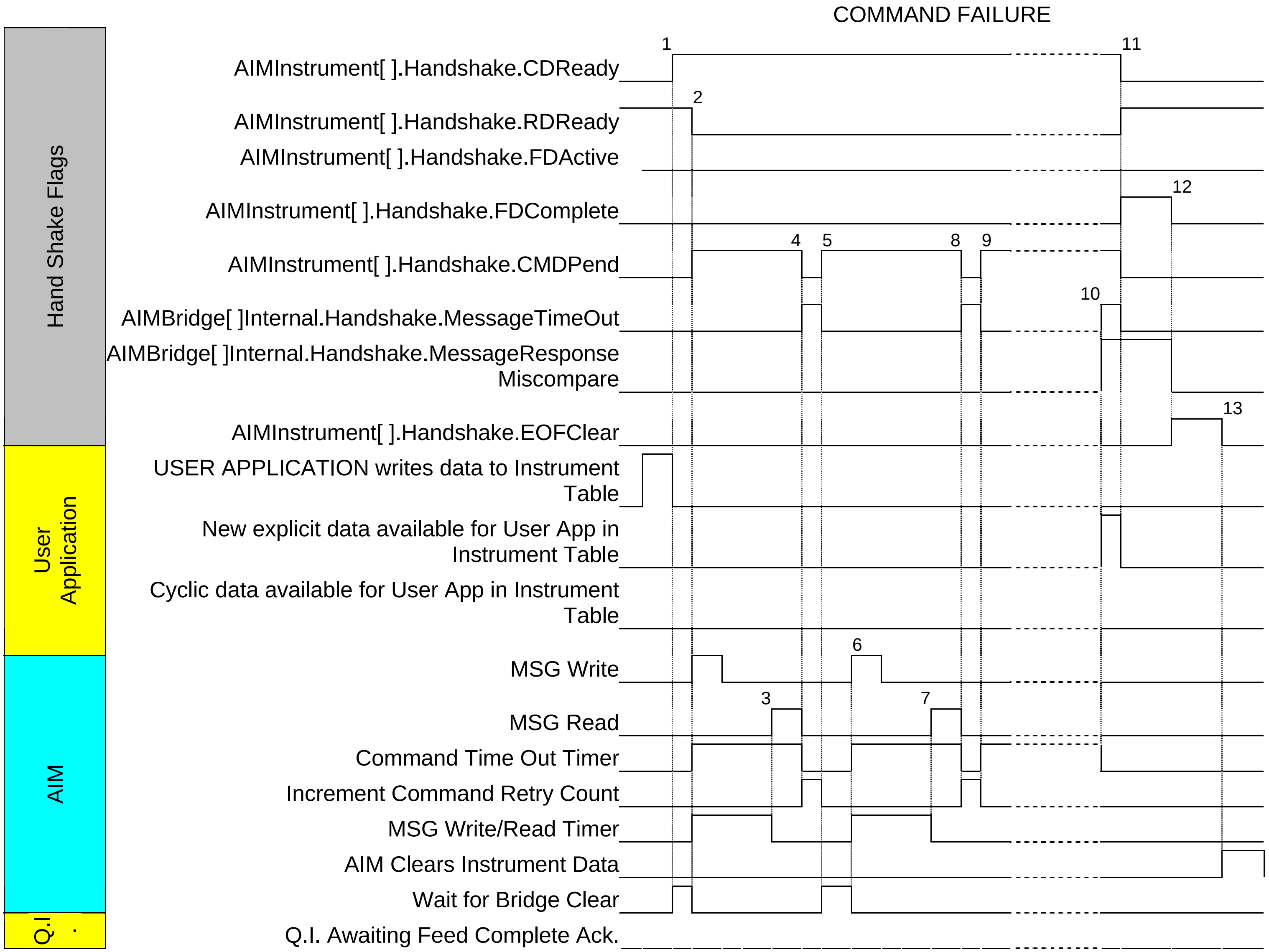

COMMAND FAILURE
Hand Shake Flags
User Application
AIM
Q.I.
AIMInstrument[ ].Handshake.CDReady
AIMInstrument[ ].Handshake.RDReady
AIMInstrument[ ].Handshake.FDActive
AIMInstrument[ ].Handshake.FDComplete
AIMInstrument[ ].Handshake.CMDPend
AIMBridge[ ]Internal.Handshake.MessageTimeOut
AIMBridge[ ]Internal.Handshake.MessageResponse Miscompare
AIMInstrument[ ].Handshake.EOFClear
USER APPLICATION writes data to Instrument Table
New explicit data available for User App in Instrument Table
Cyclic data available for User App in Instrument Table
MSG Write
MSG Read
Command Time Out Timer
Increment Command Retry Count
MSG Write/Read Timer
AIM Clears Instrument Data
Wait for Bridge Clear
Q.I. Awaiting Feed Complete Ack.
1
2
3
4
5
6
7
8
9
10
11
12
13

Figure 23: Typical Failure

(Reference numbers within the figure refer to steps in the previous table.)

4.2.3 Command Processing Overview

Once the user's application initiates the command to begin a transfer, there is no further intervention required to complete a feed under normal circumstances.

The following flow chart depicts the processing of data.

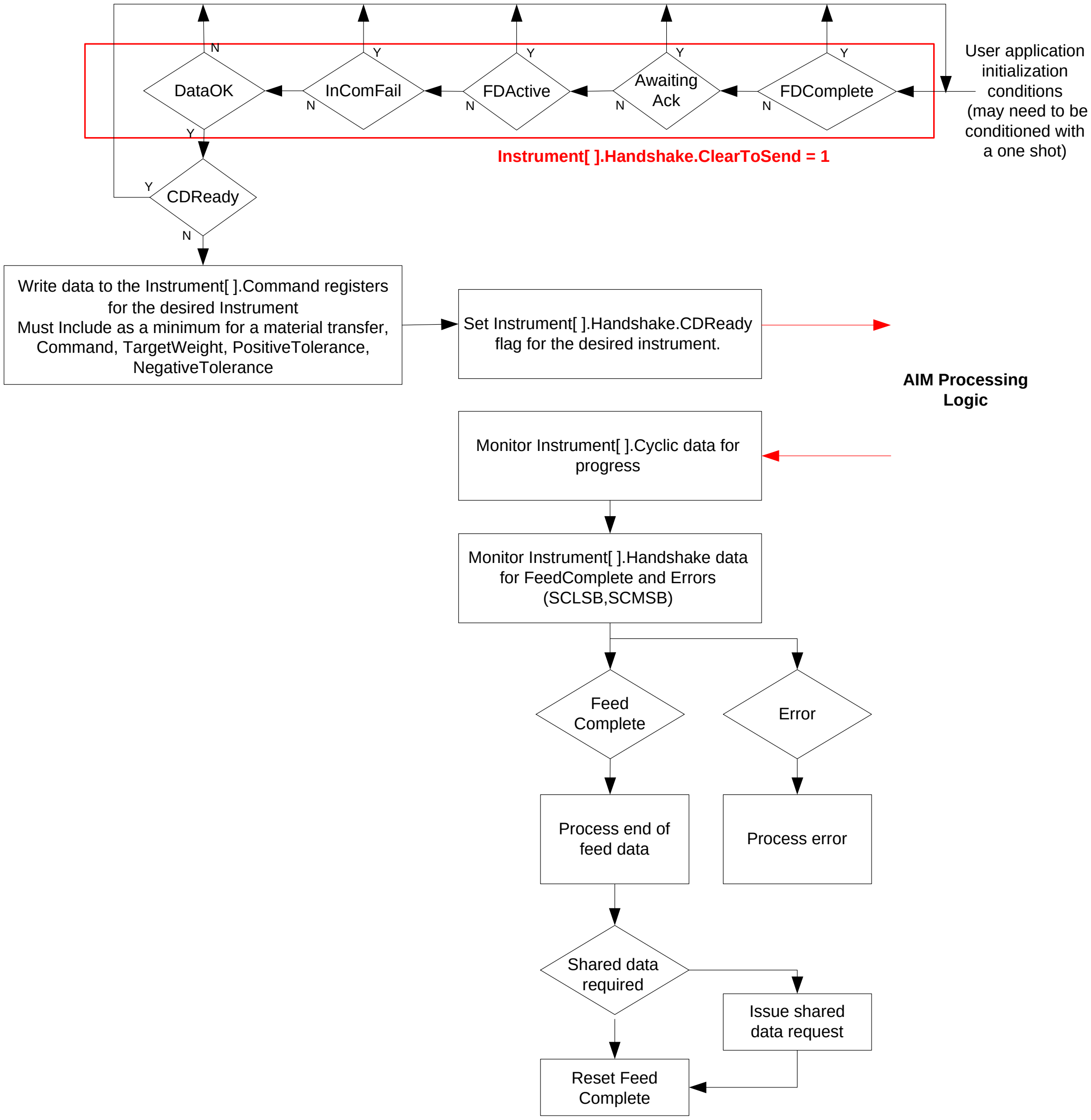

User application initialization conditions (may need to be conditioned with a one shot)
FDComplete
Awaiting Ack
FDActive
InComFail
DataOK
Y
N
Instrument[ ].Handshake.ClearToSend = 1
CDReady
Write data to the Instrument[ ].Command registers for the desired Instrument Must Include as a minimum for a material transfer, Command, TargetWeight, PositiveTolerance, NegativeTolerance
Set Instrument[ ].Handshake.CDReady flag for the desired instrument.
AIM Processing Logic
Monitor Instrument[ ].Cyclic data for progress
Monitor Instrument[ ].Handshake data for FeedComplete and Errors (SCLSB,SCMSB)
Feed Complete
Error
Process end of feed data
Process error
Shared data required
Issue shared data request
Reset Feed Complete

Figure 24: Command Processing – User Application

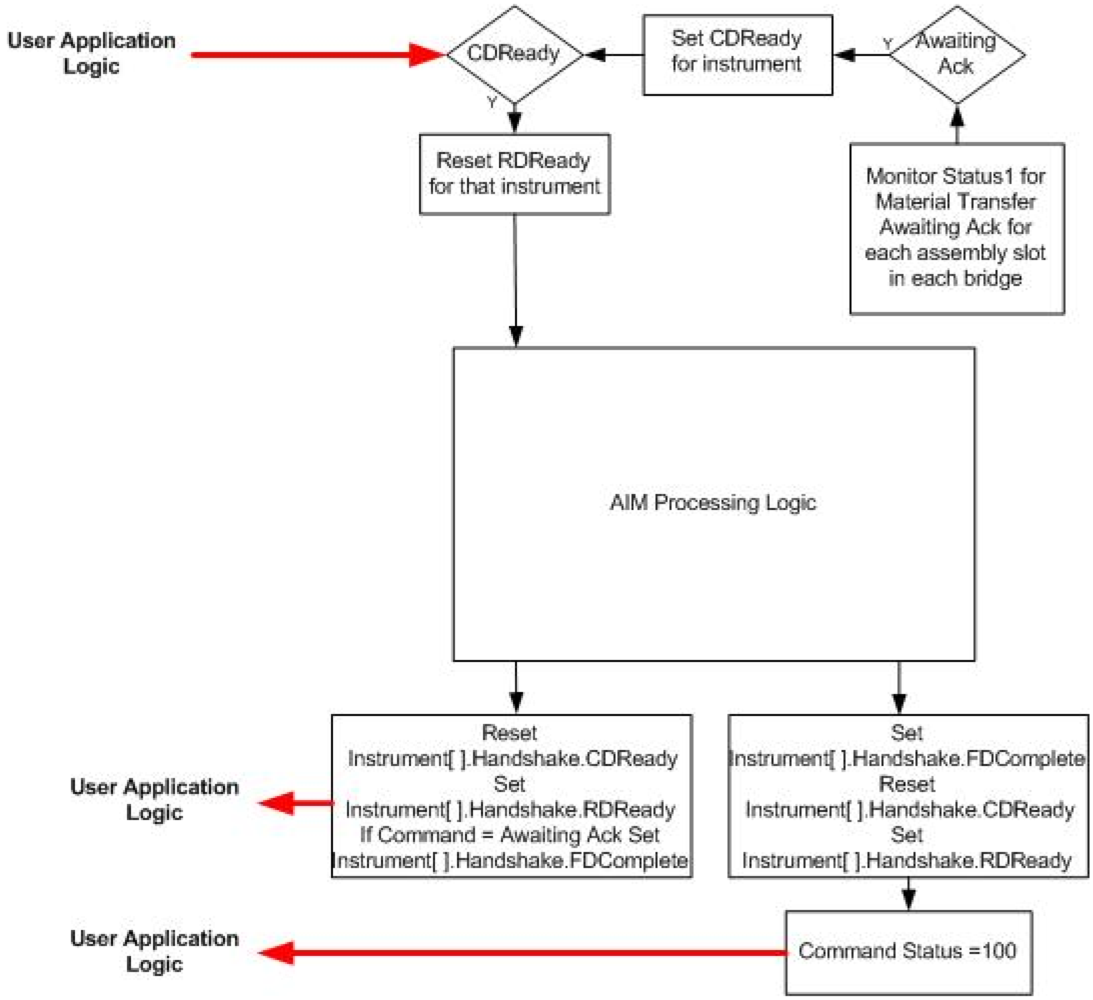

User Application Logic
CDReady
Y
Set CDReady for instrument
Y
Awaiting Ack
Reset RDReady for that instrument
Monitor Status1 for Material Transfer Awaiting Ack for each assembly slot in each bridge
AIM Processing Logic
Reset Instrument[ ].Handshake.CDReady Set Instrument[ ].Handshake.RDReady If Command = Awaiting Ack Set Instrument[ ].Handshake.FDComplete
Set Instrument[ ].Handshake.FDComplete Reset Instrument[ ].Handshake.CDReady Set Instrument[ ].Handshake.RDReady
User Application Logic
Command Status =100
User Application Logic

Figure 25: Command Processing - AIM

4.2.4 Error Handling

The AIM performs the following error handling tasks while processing command data.

4.2.4.1 Verifying Commands

Upon detecting a response message, the AIM compares the message response channel, sequence number, material path index and command number with those of the command message sent to that instrument via the Bridge. If they match, the response is valid and the command status and other pertinent information are copied to the Instrument Data structure.

If the Command Response does not match the Command, the command is issued a second time - provided the configured number of retries is greater than one.

Note: The sequence number is not incremented during retries, although under normal circumstances the AIM does generate a unique sequence number each time a command is issued.

The command is re-issued until the retry count equals the configured number of retries. At this point the Message Response Miscompare (AIMBridge[ ].Internal.Hanshake.Miscompare)MessageResponseMiscompare error is generated within the Bridge.

Upon generating an AIMBridge[ ].Internal.Handshaking.MessageResponseMiscompareare error within the Bridge, the AIM generates a command status of 100 and sends this command status to the Instrument Data Structure, AIMInstrument[ ].Response.CommandStatus. If the Feed Active indication (AIMInstrument[ ].Cyclic.Status.CycleActive) ,within the instruments cyclic data, is not on when the MessageResponseMiscompareare occurs, the AIM presumes that the feed was not initiated within the Q.iMPACT Matroller. If this is the case the AIM generates a Feed Complete indication (AIMInstrument[ ].Handshake.FDComplete), writes the negative of the target weight from the command (AIMInstrument[ ].Command.TargetWeight) to the target error in the response data (AIMInstrument[ ].Reponse.TargetError), writes the MPI from the command (AIMInstrument[ ].Command.MaterialPath) to the Active MPI (AIMInstrument[ ].Response.MaterialPath), and sets the under tolerance material transfer status qualifier (AIMInstrument[ ].Response.MagerialTransferStatusQualifier) to a value of 1 (underweight). This effectively indicates a material transfer of zero (0), since according to the Q.iMPACT Matroller, the material transfer never took place.

If the Feed Active indication for a particular instrument is on when the AIMBridge[ ].Internal.Handshaking.MessageResponseMiscompareare error is generated, the AIM writes the MPI from the command (AIMInstrument[ ].Command.MaterialPath) to the active MPI (AIMInstrument[ ].Response.MaterialPath) and returns a command status (AIMInstrument[ ].Reponse.CommandStatus of 100. In this case the AIM presumes that the command did initiate the material transfer within the Q.iMPACT Matroller, although the communications handshaking with the Q.iMPACT Matroller did not complete as expected.

4.2.4.2 Timeouts

The AIM monitors the status of each (ControlNet) message write and message read command to ensure that they complete within a preset amount of time. The nature of the explicit messaging dictates that commands sent to a Q.iMPACT Matroller are sent in pairs, a write followed by a read. If the message pairing does not complete within a preset time, a timeout is declared, the AIMBridge[ ].Internal.Handshake.MessageTimeOut flag is set, and the AIMBridge[ ].Internal.RetryCount is incremented.

If the configured number of retries AIMBridge[ ].Configuration.Number of retries is greater that 1, then the message is resent. This resending continues each time a retry is executed. Each time the timer preset is exceeded a Message Time Out (AIMBridge[ ].Internal.Handshake.MessageTimeOut) error is generated and reported in the handshaking word of the corresponding Bridge.

4.2.4.3 Command Level Errors

Command level errors, errors that are returned from the Q.iMPACT Matroller (SeeAIMInstrument[ ].Response.CommandStatus), must be handled by the User Application.

Status Class and Status Type define the Command Status returned from the Q.iMPACT Matroller.

There are three Status Classes:

1. Success
2. Warning
3. Error

Within the ERROR Status Class there are four Status Types.

1. Bad Data
2. Communications
3. Instrument
4. Process Error

More detail on these errors is included in the troubleshooting section.

Upon receiving a command level error, the AIM sets the AIMInstrument[ ].Handshake.FDComplete flag. The material feed data becomes available within the Instrument Data Structure, for the User Application to monitor.

4.2.4.4 Unexpected Termination of Material Transfer

If a Material Transfer is active at the point when a Master/Instrument reset is issued to a particular Q.iMPACT Matroller, the Material Transfer is immediately terminated within the Q.iMPACT Matroller. At this point the Matroller does not set the Awaiting Acknowledgement flag (AIMInstrument[ ].Cyclic.Status.AwaitingAck). The AIM monitors for this condition, and once detected, the AIM sets the AIMInstrument[ ].Handshake.FDComplete indicator (Feed Complete), sets the command status (AIMInstrument[ ].Response.CommandStatus) to 101 (Error - Material Transfer Ended Without Request For Acknowledge), and sets the Material Transfer Status Qualifier to the Under Tolerance state. During a Material Transfer the AIM buffers the AIMInstrument[ ].Cyclic.FeedWeight, if for some reason the Material Transfer terminates unexpectedly the AIM writes this buffered value to the Instrument.Response.FeedWeight, and calculates the error (AIMInstrument[ ].Response.TargetError) using the initial target weight setpoint AIMInstrument[ ].Command.TargetWeight.

4.2.4.5 Reset of FDComplete

The FDComplete indicator (AIMInstrument[ ].Handshake.FDComplete) notifies the User Application that a feed is complete. If a Material Transfer executes during a communications error between the PLC and the Q.iMPACT Matroller, an erroneous Feed Complete indication may exist after the Material Transfer is complete. It is also possible that the User Application will use the Feed Complete indication as an interlock within its logic.

Therefore, the AIM contains logic to reset the FDComplete indicator - if it remains on for a pre-configured length of time. This function resets the FDComplete indication when the User Application may be locked up and cannot reset. This feature can be enabled or disabled in the Configuration.

When a material transfer begins, the AIM buffers the amount of fed data from the Cyclic Data table. If at any point the Feed Active (AIMInstrument[ ].Cyclic.Status.CycleActive)) transitions to a low state for a set period of time (AIMInstrument[ ].Configuration.FeedNotActiveDebounceTimerPreset) without an Awaiting Acknowledge indication from the Q.iMPACT Matroller, the AIM treats this as an Unexpected Termination of Material Transfer.

4.2.4.6 ControlNet Reset

The AIM sends data to the Q.iMPACT Matroller via ControlNet explicit messages. The AIM then monitors the status of the explicit ControlNet messages. If the number of explicit error messages exceeds the pre-configured threshold, the AIM sends a ControlNet reset command (AIMBridge[ ].Internal.Handshake.CnetResetActive) to the corresponding Bridge.

The AIM then waits for a preset length (AIMBridge[ ].Configuration.ControlNetResetTimerPreset) of time before issuing another command to the Q.iMPACT Matroller. During this wait period, the AIM sets an Awaiting ControlNet reset flag (AIMBridge[ ].Internal.Handshake.AwaitingCnetReset).

4.2.4.7 Null Commands

The Q.iMPACT Matroller ignores erroneous command data. The AIM, however, performs certain limit checks on the command data. If the command data does not fall within the defined limits then the command will be ignored.

The AIMInstrument[ ].Handshake.CDReady flag is reset within the instrument data structure.

The defined command data limits are as follows:

AIMInstrument[ ].Command.MaterialPath > 0
AIMInstrument[ ].Command.Command >0

5 Configuration

The AIM has a number of configuration variables that must be set up to reflect the application. These variables provide flexibility and ease of operation.

Each of the three main data structures within the AIM has configuration items.

1. AIM.Configuration Items
2. Bridge.Configuration Items
3. Instrument.Configuration Items

There are two key configuration items, **which must be set first**. These two items are:

1. AIM.Configuration.NumberofBridges
2. AIMBridge[X].Configuration.NumberofInstruments *

***** This variable must be configured for each Bridge.

5.1 AIM Configuration Items

The following AIM level items must be configured in the AIM configuration data structure.

5.1.1 AIM.Configuration.CyclicDataFailureTimerPreset

Each active instrument within a Q.iMPACT cluster has a watchdog handshake bit which alternates states at a given rate, and is retrieved by the Controller via Cyclic data (AIMInstrument[ ].Cyclic.STATUS 1.ComIntegrity). The purpose of this flag, within the Q.iMPACT architecture, is to report the proper operation of an instrument within a given assembly slot.

The AIMInstrument[ ].Internal.CyclicDataFailuretimer monitors the length of time between low states of the CommIntegrity flag within STATUS. Once this timer has timed out a communications error is generated (AIMInstrument[ ].Handshake.InComFail).

The length of time between Cyclic data reads is variable and depends on the ControlNet network traffic. The preset value of this timer may be configured using the AIMInstrument[ ].Configuration.CyclicDataFailureTimerPreset variable.

Note: Each instrument has its own Cyclic Data Failure Timer. The AIM.Configuration.CyclicDataFailureTimerPreset is used as the global preset for all Instruments if AIM.Configuration.EnableGlobalPresets is activated.

The timer preset value should be set no lower than 500 (500ms). The default value shipped with the software is 15000 (15 seconds).

Table 25: Configuration Tag Summary

Tag	Description	Default
AIM.Configuration.CyclicDataFailureTimerPreset	Instrument Communications Timer Preset	15000 (Tbase=.001)

5.1.2 AIMConfiguration.FeedCompleteResetTimerPreset

This is the Feed Complete Reset Timer Preset location. If the Automatic Feed Complete Reset control is enabled, this value is used as the timer preset. It will time the reset for the Feed Complete. If the Feed Complete is on for longer than the timer preset, the AIM will reset the Feed Complete.

Note: Each instrument has its own Feed Complete Reset Timer. The AIM.Configuration.FeedCompleteResetTimerPreset is used as the global preset for all Instruments if AIM.Configuration.EnableGlobalPresets is activated.

The timer preset value should be set no lower than 500 (500ms). The default value shipped with the software is 5000 (5 seconds).

Table 26: Configuration Word Summary

Tag	Description	Default
AIM.Configuration.FeedCompleteResetTimerPreset	Feed Complete Reset Timer Preset	5000 (Tbase=.001)

5.1.3 AIM.Configuration.FeedNotActiveTimerPreset

This is the Feed Not Active Timer Preset location. In the event that Cyclic Data is temporarily interrupted, the Feed Active indication may go low for an active Material Transfer. The Feed Active indication low condition is used to initiate a timer, which acts as a debounce for the Feed Active low signal.

During the course of a Material Transfer the Feed Active signal may be set low if a Master/Instrument reset is issued to the Q.iMPACT Matroller. The debounce time is used to distinguish between a temporary loss of the Feed Active signal during a Material Transfer and the loss of the Feed Active signal due to reset.

Note: Each instrument has its own Feed Not Active Timer. The AIM.Configuration.FeedNotActiveTimerPreset is used as the global preset for all Instruments if AIM.Configuration.EnableGlobalPresets is activated.

The timer preset value should be set no lower than 500 (500ms). The default value shipped with the software is 1000 (1 second).

Table 27: Configuration Word Summary

Tag	Description	Default
AIM.Configuration.FeedNotActiveTimerPreset	Feed Not Active Timer Preset	1000 (Tbase=.001)

5.1.4 AIM.Configuration.CNetResetErrorCountPreset

This is the ControlNet Reset Error Count Preset location. During the course of communications with a Q.iMPACT Matroller, the communications may result in a ControlNet error.

If the PLC detects the number of errors configured in this parameter within a one hour period, a ControlNet reset command is issued automatically by the AIM to the Q.iMPACT Matroller, if the AIM.Configuration.AutoCNetResetEnable control is set.

Table 28: Configuration Word Summary

Tag	Description	Default
AIM.Configuration.CNetResetErrorCountPreset	ControlNet Reset Error Count Preset	4

5.1.5 AIM.Configuration.NumberofBridges

This configuration item defines the number of Bridges in the application. The AIM is designed to support communications with ten bridges, which is the maximum number of Bridges allowed in a Q.iMPACT Cluster. This configuration item defines the number of Bridges that the AIM will process.

Note: The user must size the Bridge tag's array to suit the number of Bridges defined by this configuration item; otherwise a Controller fault will occur.

Table 29: Configuration Word Summary

Tag	Description	Default
AIM.Configuration.NumberofBridges	Configured number of bridges within the Q.iMPACT cluster	1

5.1.6 AIM.Configuration Function Enable/Disable Bits

To enhance the AIM's flexibility, certain sections of code and/or data processing may be enabled or disabled by the controls in the AIM.Configuration data structure.

Table 30: AIM.Configuration Data Structure Members

Tag	.Members	Description	Default
AIM.Configuration			
	.MATIDEnable	Enable Material Transfer ID processing. When this address is in the zero state, this control disables the processing if Material ID data. When set, in the on state, it allows the Material ID data to be processed	0 = Off (Disabled)
	.Reset	Reset AIM. When this control is set it: Clears all data from the Instrument Data Table Resets all AIM pointers Resets all Handshake bits Clears all data from the Bridge Table Clears the Command Write Buffer Clears the Command Response Buffer	0 = Off (Disabled)

Tag	.Members	Description	Default
	.AutoFeedCompleteResetEnable	Automatic Feed Complete Reset Enable (Exception Handling) If the Automatic Feed Complete Reset control is enabled and the Feed Complete is on longer than the timer preset, the AIM will reset the Feed Complete based on the assumption that the controlling User Application is not going to perform this operation.	0 = Off (Enabled)
	.AutoCNetResetEnable	ControlNet Automatic Reset Enable/Disable. The AIM can automatically reset the ControlNet when the preset number of ControlNet communications errors occur.	0 = Off (Enabled)
	.BridgeDiagnosticCountResetEnable	Diagnostic Count Reset Each Hour Enable/Disable. The AIM can automatically reset the diagnostic counts each hour	0 = Off (Enabled)
	.CyclicDataCheckSumEnable	Cyclic Data Check Sum Validation Enable/Disable. During Cyclic Data communications the Q.iMPACT Matroller generates a Cyclic Data check sum. The AIM independently generates a check sum from the data received in the cyclic data transmission. The check sum calculated by the Q.iMPACT Matroller is compared to the check sum calculated in the AIM. If the two are not equal, the AIM suspends operation until the check sum comparison is true again.	0 = Off (Disabled)
	.SimulationEnable	Simulation Enable/Disable. The AIM can provide instrument simulation. During simulation all ControlNet commands are disabled. This control switches the AIM between its normal operating mode and simulation mode.	0 = Off (Disabled)
	.EnableGlobalPresets	Three timer presets may be individually set via the Instrument configuration data, or all instruments may be set to the global configuration values within the AIM configuration data. This control either enables or disables the global timer presets.	0 = Off (Enabled)
	.ClearDataonFDCompleteReset	Enable Clear Instrument data Upon Feed Complete Reset. This control allows the data within the Instrument Data structure (Command/Response) to be cleared when the Feed Complete Bit is reset. A reset occurs for a particular instrument when it is low. When this control is set, the Instrument Data Table is not cleared upon a Feed Complete reset.	0 = Off (Enabled)
	.ResetToFactoryDefaults	When this control is set the AIM restores the Factory Default settings for all configuration items within the AIM application.	0 = Off (Disabled)
	.ENetCommunicationEnable	Enables the AIM to communicate to the Q.i.(s) via Ethernet	0=Off (ControlNet)

5.2 Bridge Configuration Items

The following Bridge level items must be configured in the Bridge configuration data structure.

5.2.1 AIMBridge[ ].Configuration.ControlNetResetTimerPreset

When the AIM issues a ControlNet reset command, ControlNet communications is unavailable while the communications hardware resets. This time is the duration the AIM waits, after a ControlNet reset is issued, before attempting to send commands to the Q.iMPACT Matroller.

The timer preset value should be set no lower than 30000 (30 sec). The default value shipped with the software is 60000 (60 seconds).

Table 31: Configuration Word Summary

Tag	Description	Default
AIMBridge[ ].Configuration.ControlNetResetTimerPreset	Bridge ControlNet reset Timer Preset	60000 (Tbase=.001)

5.2.2 AIMBridge[ ].Configuration.ReadWriteTimerPreset

This is the Bridge Write/Read Timer Preset location. Once a command is sent to a Q.iMPACT Matroller, there is an inherent delay within the Q.iMPACT Matroller before the Command Read information becomes available. The AIMBridge[ ].Internal.ReadWriteTimer Preset delay is used by the AIM between the MSG Write and Read instructions. Use this configuration item to set the length of the delay.

Once a MSG Write instruction is executed, the timer increments to this preset and then issues the MSG Read instruction.

Note: Mettler-Toledo does not guarantee the availability of response data in less than 250ms.

This configuration item should not be set less than 250 (250ms).

Table 32: Configuration Word Summary

Tag	Description	Default
AIMBridge[ ].Configuration.ReadWriteTimerPreset	Bridge Write/Read Timer Preset	500 (Tbase=.001)

5.2.3 AIMBridge[ ].Configuration.CommunicationsTimeOutTimerPreset

This is the Communications Handshake Time Out Timer Preset location. The handshake time out reflects the time needed to send a command to a particular Q.iMPACT Matroller and receive a valid Command Response.

Once this time has elapsed - during a particular write/read operation - a time out message occurs (AIMBridge[ ].Internal.Handshake.Message.Timeout).

Note: To process the command and supply response data, the Q.iMPACT Matroller uses a time delay for write/read operations. This time delay should not be set lower than twice the Bridge write/read time. If a value is entered that is too low, the AIM will automatically adjust the time preset to two times the Bridge write/read timer preset.

The default value shipped with the software is 1000 (1 second).

Table 33: Configuration Word Summary

Tag	Description	Default
AIMBridge[ ].Configuration.ComunicationsTimeOutTimerPreset	Communications Handshake Time Out Timer Preset	1000 (Tbase=.001)

5.2.4 AIMBridge[ ].Configuration.NumberofRetries

This is the Retry Count Preset location. The Retry Count Preset determines the number of Message Timeout errors for a particular command until a Message Response Miscompare error (AIMBridge[ ].Internal.Handshake.MessageResponseMiscompare) is generated. The minimum value for the Retry Count Preset for a particular Bridge is one.

If the retry count preset is exceeded, the Message Response Mis-compare error is generated and a Command Status of 100 is returned to the Instrument Data structure (AIMInstrument[ ].Response.CommandStatus).

Table 34: Configuration Word Summary

Tag	Description	Default
AIMBridge[ ].Configuration.NumberofRetries	Retry Count Preset	5

5.2.5 AIMBridge[ ].Configuration.NumberofInstruments

This configuration item stores the number of Instruments per Bridge that are configured. The Q.iMPACT architecture supports a maximum of 24 instruments for a given bridge, and a maximum of 200 instruments in a cluster.

Instrument data is stored in the Instrument data structure, which contains the instrument data for all Bridges in the cluster. The AIM automatically determines which group of instruments belong to a certain Bridge based on the configuration information set here. The AIM accomplishes this by calculating the first instrument index for each configured Bridge (AIMBridge[ ].Configuration.FirstInstrument).

Table 35: Configuration Word Summary

Tag	Description	Default
AIMBridge[ ].Configuration.NumberofInstruments	Number Of Instruments for a given Bridge	4

AIMBridge[ ].Configuration.Calculated.FirstInstrument
The AIM calculates the first instrument index for each configured Bridge this value is termed AIMBridge[ ].Configuration.FirstInstrument.

The following example shows how to set up these registers for a three Bridge system with 12 instruments on Bridge#1, seven on Bridge#2, and nine on Bridge#3.

Table 36: Example

Configuration Item	Configured Value
AIM.Configuration.NumberofBridges	3
AIMBridge[1].Configuration.NumberofInstruments	12
AIMBridge[2].Configuration.NumberofInstruments	7
AIMBridge[3].Configuration.NumberofInstruments	9

Calculated Variable	Calculation	Calculated Value
AIMBridge[1].Internal.CalculatedFirstInstrument	1	1
AIMBridge[2].Internal.CalculatedFirstInstrument	1+ AIMBridge[1].Configuration.NumberofInstruments	13
AIMBridge[3].Internal.CalculatedFirstInstrument	1+ AIMBridge[1].Configuration.NumberofInstruments + AIMBridge[2].Configuration.NumberofInstruments	20

Note: The Instrument tag’s array must be configured to the correct number of instruments otherwise a controller fault will occur. The array should be sized for the total number of instruments used by all defined Bridges +1. Using the example above the size of this array would be 29 (12+7+9+1).

Note: When designing a system, take care when defining the number of instruments. A good design practice is to leave a certain number of spare instruments within the defined number of instruments, which allows for future expansion. The processing time and controller memory required for spare instruments is negligible. A good guideline would be 25-50% spare instruments, depending on the project and the probability of expansion.

5.2.6 AIMBridge[ ].Configuration.ENetCyclicReadTimerPreset

This timer preset represents the time between controller messages used in gathering cyclic Q.i. data when implementing Ethernet communications. This variable needs to be defined for each active Q.i. bridge.

		Default
		2000 (Tbase=0.001)

5.3 Instrument Configuration Items

The following Instrument configuration items must be set up.

5.3.1 AIMInstrument[ ].Configuration.FeedCompleteResetTimerPreset

This is the Feed Complete Reset Timer Preset location. If the Automatic Feed Complete Reset control is enabled, this value is used as the timer preset. It will time the reset for the Feed Complete. If the Feed Complete is on for longer than the timer preset, the AIM will reset the Feed Complete.

Note: Each instrument has its own Feed Complete Reset Timer. The AIM.Configuration.FeedCompleteResetTimerPreset is used as the global preset for all Instruments if AIM.Configuration.EnableGlobalPresets is activated.

The timer preset value should be set no lower than 5000 (5 seconds). The default value shipped with the software is 5000 (5 seconds).

Table 37: Configuration Word Summary

Tag	Description	Default
AIMInstrument[ ].Configuration.FeedCompleteResetTimerPreset	Feed Complete Reset Timer Preset	5000 (Tbase=.001)

5.3.2 AIMInstrument[ ].Configuration.FeedNotActiveDebounceTimerPreset

This is the Feed Not Active Timer Preset location. If the Cyclic Data is temporarily interrupted, the Feed Active indication may go low for an active Material Transfer. The Feed Active indication low condition is used to initiate a timer, which acts as a debounce for the Feed Active low signal.

During the course of a Material Transfer the Feed Active signal may be set low if a Master/Instrument reset is issued to the Q.iMPACT Matroller. The debounce time is used to distinguish between a temporary loss of the Feed Active signal during a Material Transfer and the loss of the Feed Active signal due to reset.

Note: Each instrument has its own Feed Not Active Timer. The AIM.Configuration.FeedNotActiveTimerPreset is used as the global preset for all Instruments if AIM.Configuration.EnableGlobalPresets is activated.

The timer preset value should be set no lower than 500 (500ms). The default value shipped with the software is 1000 (1sec).

Table 38: Configuration Word Summary

Tag	Description	Default
AIMInstrument[ ].Configuration.FeedNotActiveDebounceTimerPreset	Feed Not Active Timer Preset	1000 (Tbase=.001)

5.3.3 AIMInstrument[ ].Configuration.CyclicDataFailureTimerPreset

Each active instrument within a Q.iMPACT cluster has a watchdog handshake bit, which alternates states at a given rate, and is retrieved by the Controller via Cyclic data (AIMInstrument[ ].Cyclic.STATUS 1.ComIntegrity). This flag within the Q.iMPACT architecture reports the proper operation of an instrument within a given assembly slot.

The AIMInstrument[ ].Internal.CyclicDataFailureTimer monitors the length of time between low states of the CommIntegrity flag within STATUS. If this timer times out, a communications error is generated (AIMInstrument[ ].Handshake.InComFail).
The length of time between Cyclic data reads is variable and depends on the ControlNet network traffic. For this reason, the preset value of this timer is configurable using the AIMInstrument[ ].Configuration.CyclicDataFailureTimerPreset variable.

Note: Each instrument has its own Cyclic Data Failure Timer. The AIM.Configuration.CyclicDataFailureTimerPreset is used as the global preset for all Instruments if AIM.Configuration.EnableGlobalPresets is activated.

The timer preset value should be set no lower than 10000 (10 seconds). The default value shipped with the software is 15000 (15 seconds).

Table 39: Configuration Tag Summary

Tag	Description	Default
AIMInstrument[ ].Configuration.CyclicDataFailureTimerPreset	Instrument Communications Timer Preset	15000 (Tbase=.001)

5.3.4 AIMInstrument[ ].Configuration.ACKHandshakeTimerPreset

If for some reason the Acknowledge sent by the AIM (automatically generated) fails, this timer will time out when the Accumulated = Preset, and the acknowledge command will be re-issued. If the acknowledge command does not fail, then this timer is not used.

Note: Each instrument has its own Configuration .ACKHandshake Timer. The AIMInstrument[ ].Configuration.ACKHandshakeTimerPreset is used as the global preset for all Instruments if AIM.Configuration.EnableGlobalPresets is activated.

Table 40: Configuration Tag Summary

Tag	Description	Default
AIMInstrument[ ].Configuration.ACKHandshakeTimerPreset	Instrument Acknowledge Handshake Timer Preset	5000 (Tbase=0.001)

5.4 Factory Defaults

The AIM ships with default values within the various configuration variables. These default values are also stored within the AIM’s Factory Default data structure. At any time you can enable the AIM.Configuration.ResetToFactoryDefaults control to restore the default values.

The following is a list of Factory Defaults, which are stored within the AIM.

Table 41: AIM Factory Default Items

Tags	.Members	Default Setting
.FactoryDefault		
.AIM		
*	.CyclicDataFailureTimerPreset	15000 (15 Seconds)
*	.FeedCompleteResetTimerPreset	5000 (5 Seconds)
*	.ACKHandshakeTimerPreset	5000 (5 Second)
	.ControlNetErrorCountPreset	4
	.NumberofBridges	1
	.MATIDEnable	0 = Disabled
	.Reset	0 = Disabled
	.AutoFeedCompleteResetEnable	0 = Enabled
	.AutoCNetResetEnable	0 = Enabled
	.BridgeDiagnosticCountResetEnable	0 = Enabled
	.CyclicDataCheckSumEnable	0 = Disabled
	.SimulationEnable	0 = Disabled
	.EnableGlobalPresets	0 = Enabled
	.ClearDataonFDCompleteReset	0 = Enabled
	.ResetToFactoryDefaults	0 = Disabled
	.ENetCommunicationEnable	0 = ControlNet
.Bridge		
	.ControlNetResetTimer Preset	60000
	.ReadWriteTimerPreset	500 (500 ms)
	.Communications.TimeOutTimerPreset	1000 (1 Second)
	.NumberofRetries	5
	.NumberOfInstruments	4

* These default values are written to each instrument if the AIM.Configuration.EnableGlobalPresets is set.

6 Q.i. Communications

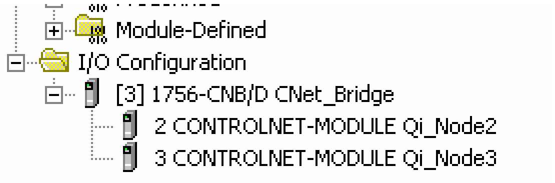
6.1 ControlNet

This section discusses the ControlNet configuration requirements. Scheduled messaging to the Cyclic Data Structure must be setup using RSLogix 5000, and configured within the AIM program. Event based messaging (explicit) between the Command Write Buffer and Command Read Buffer must be configured within the AIM program.

6.1.1 Cyclic Data Messaging

Cyclic data is setup within the software I/O configuration area of the user application using RSLogix 5000 programming software. Once setup, this data is continuously sent from the Q.iMPACT Matroller to the Controller. Data is now sent on a scheduled basis from the Q.i. Matroller to the ControlLogix processor via ControlNet. See the Programming Considerations for details on the required program configuration for Cyclic Data.

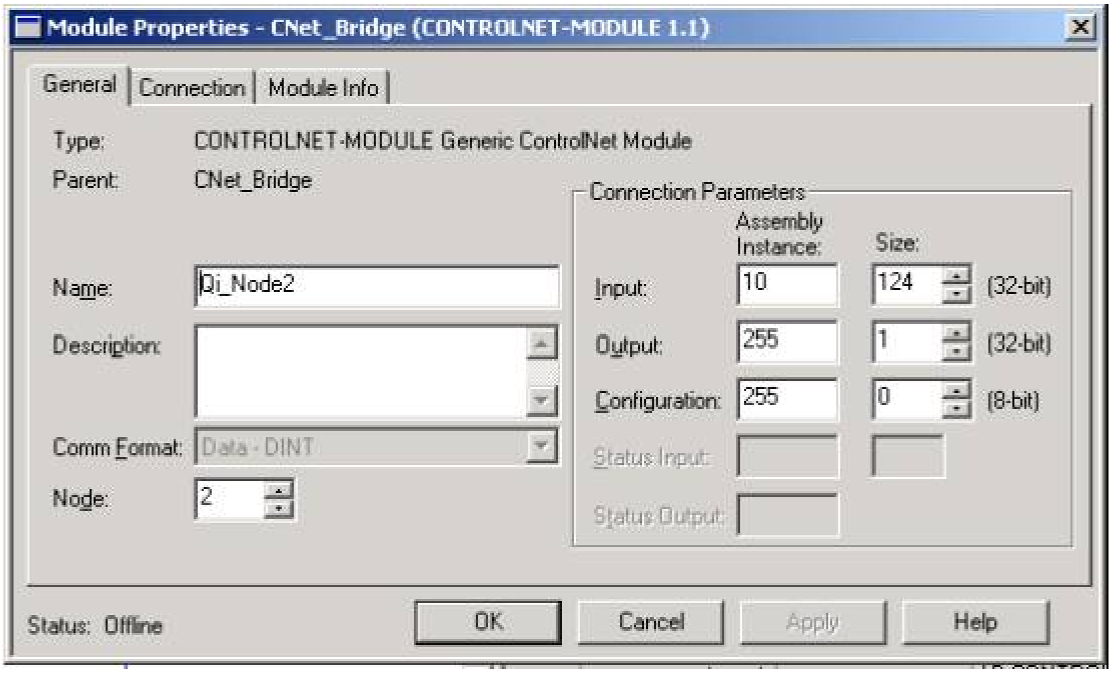
For the following example, the I/O configuration is noted.
(See the I/O configuration/settings/communications Tab in RSLogix 5000)



```
 ⊞ Module-Defined
⊟ I/O Configuration
 ⊟ [3] 1756-CNB/D CNet_Bridge
    2 CONTROLNET-MODULE Qi_Node2
    3 CONTROLNET-MODULE Qi_Node3
```

Figure 26: Control Net Module for Bridges

The following figure shows the parameter configuration for a single Q.iMPACT Matroller (single Bridge) setup to report Cyclic data to the default Bridge 1 (ControlNet Node 2 in this case). Data received from Bridge 1 Qi_Node2 is copied into AIMBridge[1].Buffer.Cyclic.



Type: CONTROLNET-MODULE Generic ControlNet Module

Parent: CNet_Bridge

Name: Qi_Node2

Description:

Comm Format: Data - DINT

Node: 2

Connection Parameters

	Assembly Instance:	Size:	
Input:	10	124	(32-bit)
Output:	255	1	(32-bit)
Configuration:	255	0	(8-bit)
Status Input:			
Status Output:			

Status: Offline

Figure 27: Cyclic Data Configuration using ControlLogix 5000

6.1.2 Explicit Messaging: Command Write Buffer

The Command Write Buffer is sent on an event basis to the Q.iMPACT Matroller. This message is sent to the Q.iMPACT Matroller via the MSG write instruction. The MSG write instructions are contained in PLC program sub-routine AIMMSGWrite.

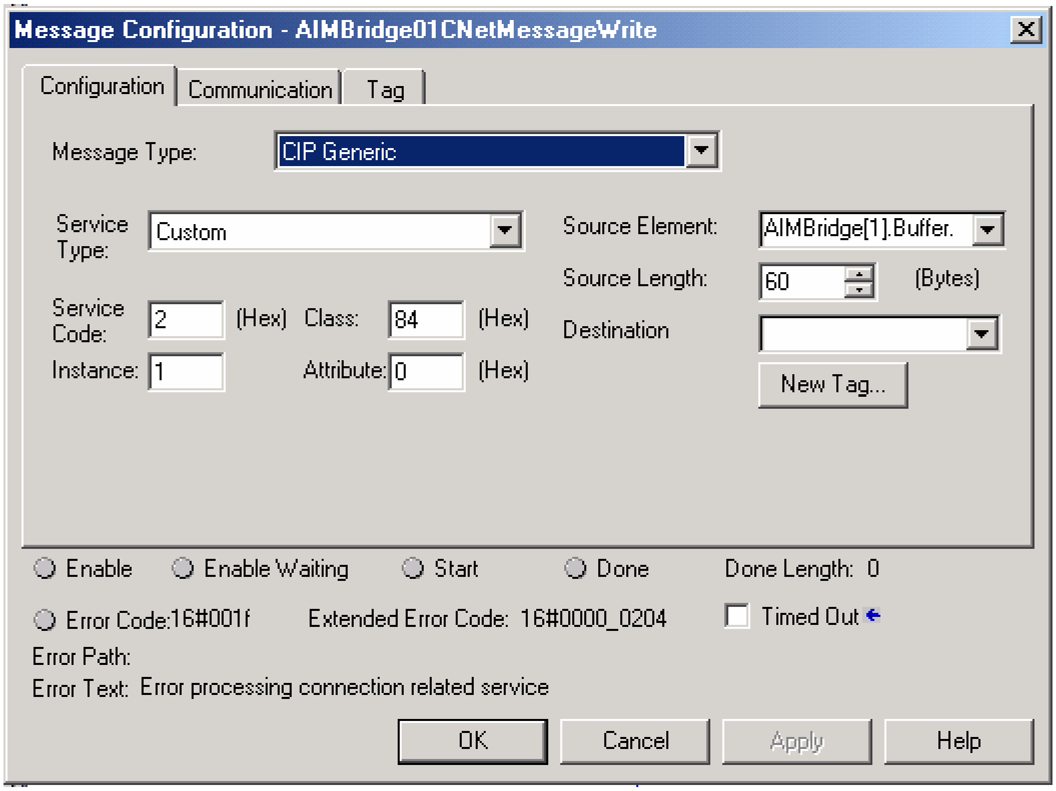
The AIM PLC code includes MSG write instructions for ten bridges.

Each MSG instruction must be edited to ensure that the configuration is consistent with the system design.

The settings for the MSG Write command are as follows:

Table 42: MSG Write Command Settings

Field	Setting
Communications Command	CIP Generic
Service Code (Hex)	2
Class Number (Hex)	84
Instance Number	1
Attribute Number	Blank
Source/Tag	AIMBridge[Bridge#].Buffer.Write
Size in Elements	60
Destination	Blank



Message Configuration - AIMBridge01CNetMessageWrite

Message Type: CIP Generic

Service Type: Custom

Service Code: 2 (Hex) Class: 84 (Hex)

Instance: 1 Attribute: 0 (Hex)

Source Element: AIMBridge[1].Buffer.

Source Length: 60 (Bytes)

Destination

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code:16#001f Extended Error Code: 16#0000_0204 □ Timed Out

Error Path:

Error Text: Error processing connection related service

Figure 28: MSG Write Command Settings

6.1.3 Explicit Messaging: Command Read Buffer

The Command Read Buffer is updated on an event basis from the Q.iMPACT Matroller. Status responses are received from the Q.iMPACT Matroller via the MSG read instruction. The MSG read instructions are contained in PLC program sub-routine AIMMSGRead.

The AIM PLC code includes MSG read instructions for ten bridges.

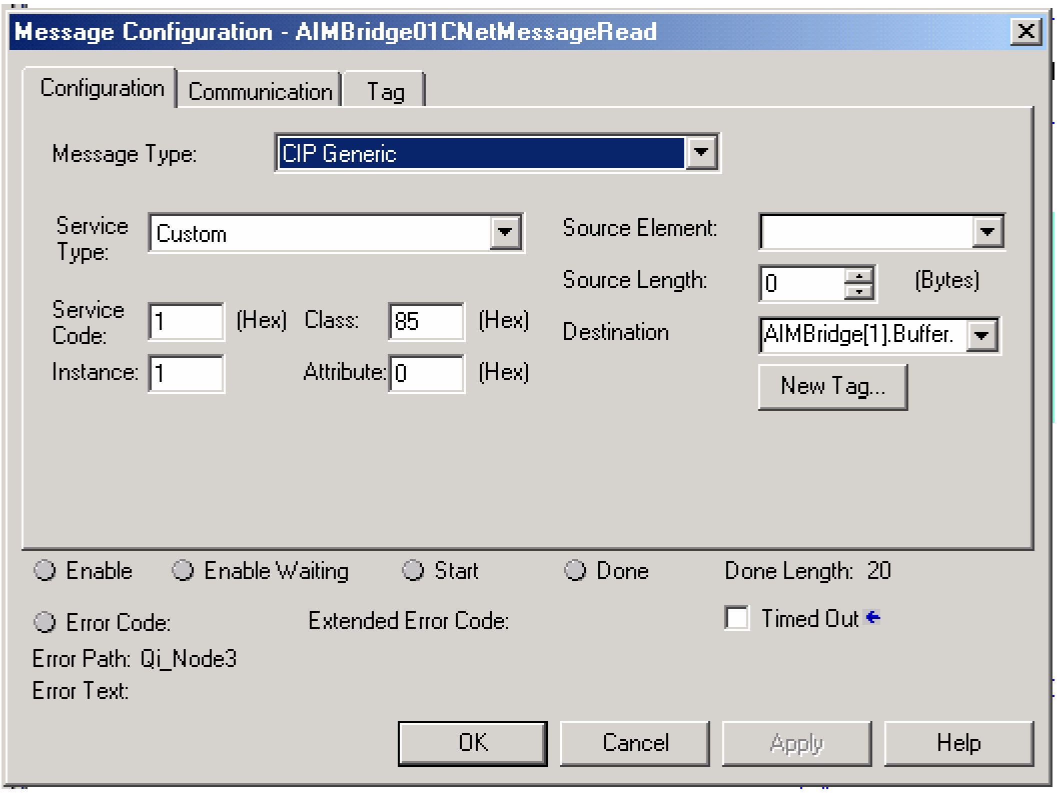
Each MSG instruction must be edited to ensure that the configuration is consistent with the system design.

The settings for the MSG Read command are as follows:

Table 43: MSG Read Command Settings

Field	Setting
Communications Command	CIP Generic
Service Code (Hex)	1
Class Number (Hex)	85
Instance Number	1
Attribute Number	Blank
Source	Blank
Size in Elements	0

Field	Setting
Destination	AIMBridge[Bridge#].Buffer.Read



Message Configuration - AIMBridge01CNetMessageRead

Configuration | Communication | Tag

Message Type: CIP Generic

Service Type: Custom

Service Code: 1 (Hex) Class: 85 (Hex)

Instance: 1 Attribute: 0 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination AIMBridge[1].Buffer.

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 20

○ Error Code: Extended Error Code: □ Timed Out

Error Path: Qi_Node3

Error Text:

Figure 29: MSG Read Command Settings

6.2 Ethernet

To implement an Ethernet solution, you must complete the two main steps detailed below:

1. Program Modifications (Configure MSG instructions)
2. AIM configuration changes

Program Modifications (Configure MSG instructions)

MSG Instruction Communications Tab

Within the AIM program, change the communications path in each of the following MSG instructions:

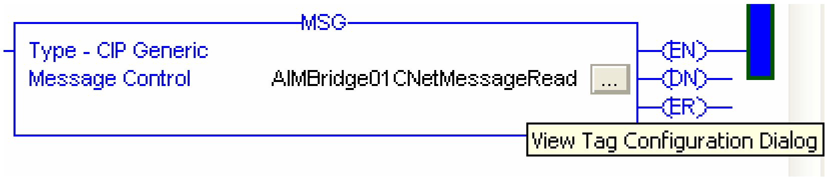
1. AIMCyclicRead" routine, instruction "AIMBridgeXXEnetCyclicRead."
2. AIMMSGRead" routine, instruction "AIMBridgeXXCnetMessageRead."
3. AIMMSGWrite" routine, instruction "AIMBridgeXXCnetMessageWrite."

The XX variables in the above communication paths represent the Bridge number for the Q.i. Matroller(s).

Note: You must modify three MSG instructions for each Bridge within the Q.i. cluster. Therefore, if a particular system has three (3) bridges, there would be a total of nine (9) MSG instructions that you would have to modify.

To change the communications path of a particular MSG instruction:

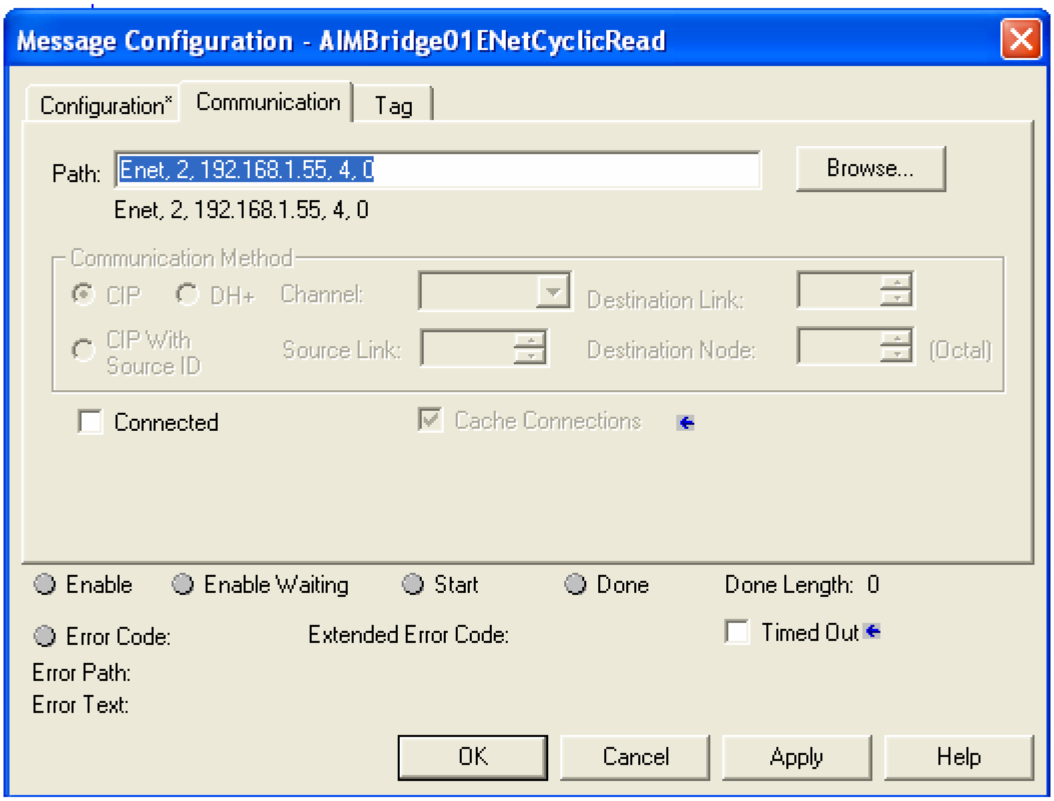
1. Go to the MSG instruction within the AIM program (routines listed above) and click on the **View Tag Configuration Dialog** button (**…**).



MSG
Type - CIP Generic
Message Control AIMBridge01CNetMessageRead ...

The **Message Configuration** window appears.

2. Select the **Communications** tab.



Message Configuration - AIMBridge01ENetCyclicRead

Path: Enet, 2, 192.168.1.55, 4, 0
Enet, 2, 192.168.1.55, 4, 0

Communication Method
● CIP ○ DH+ Channel: Destination Link:
○ CIP With Source ID Source Link: Destination Node: (Octal)

□ Connected ☑ Cache Connections

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0
○ Error Code: Extended Error Code: □ Timed Out
Error Path:
Error Text:

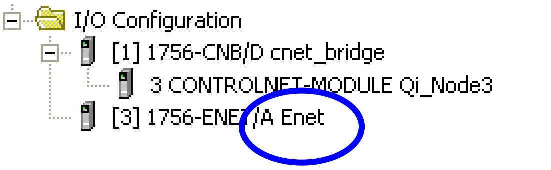
Figure 30: Message Configuration window

3. Change the Path to read “Enet, 2, XXX.XXX.XXX.XXX, 4, 0”

Where:

Enet = The name of ControlLogix Ethernet card located within the I/O configuration - see figure below.

Note: The selected Ethernet card must be physically located on the same network as the TCP/IP port (ControlNet Bridge card) of the Q.i. Matroller. Use the **Browse** button to select the card.



I/O Configuration
[1] 1756-CNB/D cnet_bridge
3 CONTROLNET-MODULE Qi_Node3
[3] 1756-ENET/A Enet

2 = Port on the ControlLogix Ethernet card.

XXX.XXX.XXX.XXX = The Ethernet IP address for the Q.i. Matroller.

Note: This is the IP address for the Ethernet on the ControlNet Bridge card. (See figure 30, Message Configuration window for the example 192.168.1.55 = IP Address of Q.i.)

4, 0 = These numbers pertain to the destination. For a Q.i, these values should always be set to these defaults.

MSG Instruction Configuration Tabs

The AIM ships with the configuration information Pre-configured and should not present any problems. If, however, the MSG communications and AIM configuration are complete and problems are experienced, verify the following information in the tables below.

Explicit Messaging: Command Write

Table 44: MSG Write Command Settings

Field	Setting
Communications Command	CIP Generic
Service Code (Hex)	2
Class Number (Hex)	84
Instance Number	1
Attribute Number	Blank
Source/Tag	AIMBridge[Bridge#].Buffer.Write
Size in Elements	60
Destination	Blank

Explicit Messaging: Command Read

Table 45: MSG Read Command Settings

Field	Setting
Communications Command	CIP Generic
Service Code (Hex)	1
Class Number (Hex)	85

Field	Setting
Instance Number	1
Attribute Number	Blank
Source	Blank
Size in Elements	0
Destination	AIMBridge[Bridge#].Buffer.Read

Explicit Messaging: Cyclic

Table 46: MSG Write Command Settings

Field	Setting
Communications Command	CIP Generic
Service Code (Hex)	1
Class Number (Hex)	86
Instance Number	1
Attribute Number	0
Source/Tag	Blank
Size in Elements	0
Destination	AIMCyclicDataBridge[Bridge#].Word

AIM Configuration Changes

Modify the following configuration settings to enable Ethernet functionality within the AIM.

1. Set AIM.Configuration.EnetCommunicationsEnable = 1
2. Set AIM.Configuration.CyclicDataCheckSumEnable = 0
3. Set AIMMBridge[x].Configuration.EnetCyclicReadTimerPreset = 1000 for each Q.i. Bridge within the Cluster.

7 Troubleshooting

This section provides troubleshooting information.

7.1 AIM Diagnostics

Within the AIM there are several diagnostic registers that may be used to assist in diagnosing problems with communications between the AIM and a Q.iMPACT Matroller.

Table 47: Bridge Diagnostic Table

Tag	.Members	Description
AIMBridge[ ].Diagnostics		
	.LastMsgWriteError	Returns the communication channels error register after each explicit write command.
	.LastMSGReadError	Returns the communication channels error register after each explicit read command.
.Current		Contains diagnostic counts for the current hour.
	.MSGWriteDoneCount	Each time a successful explicit write command is made to the Bridge this counter is incremented.
	.MSGReadDoneCount	Each time a successful explicit read command is made to the Bridge this counter is incremented.
	.CommandStatusErrorCount	Each time the Q.iMPACT returns a Command Status error (See Command Status Classifications within this document) this count is incremented.
	.CommandStatusSuccessCount	Each time the Q.iMPACT returns a Command Status success (See Command Status Classifications within this document) this count is incremented.
	.CommandStatusBusyCount	Each time the Q.iMPACT returns a Command Status busy (See Command Status Classifications within this document) this count is incremented.
	.CommandStatus100Count	Each time the AIM returns a Command Status 100 (See Command Status Classifications within this document) this count is incremented.
	.CommunicationsRetryCount	Communications retry count, current hour.
	.MSGWriteErrorCount	Each time an unsuccessful explicit write command is made to the Bridge this counter is incremented.
	.MSGReadErrorCount	Each time an unsuccessful explicit read command is made to the Bridge this counter is incremented
	.CommandStatus101Count	Each time the AIM returns a Command Status 101 (See Command Status Classifications within this document) this count is incremented.
.Previous		Contains diagnostic counts for the previous hour.
	.MSGWriteDoneCount	Each time a successful explicit write command is made to the Bridge this counter is incremented.
	.MSGReadDoneCount	Each time a successful explicit read command is made to the Bridge this counter is incremented.
	.CommandStatusErrorCount	Each time the Q.iMPACT returns a Command Status error (See Command Status Classifications within this document) this count is incremented.
	.CommandStatusSuccessCount	Each time the Q.iMPACT returns a Command Status success

Tag	.Members	Description
		(See Command Status Classifications within this document) this count is incremented.
	.CommandStatusBusyCount	Each time the Q.iMPACT returns a Command Status busy (See Command Status Classifications within this document) this count is incremented.
	.CommandStatus100Count	Each time the AIM returns a Command Status 100 (See Command Status Classifications within this document) this count is incremented.
	.CommunicationsRetryCount	Communications retry count, current hour.
	.MSGWriteErrorCount	Each time an unsuccessful explicit write command is made to the Bridge this counter is incremented.
	.MSGReadErrorCount	Each time an unsuccessful explicit read command is made to the Bridge this counter is incremented
	.CommandStatus101Count	Each time the AIM returns a Command Status 101 (See Command Status Classifications within this document) this count is incremented.

The following table lists the MSG error codes that will appear in the diagnostics registers AIMBridge[ ].Diagnostics.LastMSGReadError and AIMBridge[ ].Diagnostics.LastMSGWriteError as defined by Rockwell Automation.

Table 48: Table of MSG Error Codes

Hex Code	Error Message
0001	Connection Failure
0002	Insufficient Resource
0003	Invalid Value
0004	IOI Syntax Error
0005	Destination Unknown, Class Unsupported, Instance Undefined or structure element undefined
0006	Insufficient Packet Space
0007	Connection Lost
0008	Service Unsupported
0009	Error In data Segment
000A	Attribute List Error
000B	State Already Exist
000C	Object Model Conflict
000D	Object Already Exist
000E	Attribute Not Settable
000F	Permission Denied
0010	Device state Conflict
0011	Reply Will not fit
0012	Fragment Primitive
0013	Insufficient command data
0014	Attribute not supported
0015	Too Much Data
001A	Bridge Request too Large
001B	Bridge Response too large
001C	Attribute list shortage
001D	Invalidate attribute list
001E	Embedded service error
001F	Connection related failure
0022	Invalid reply received
0025	Key segment error
0026	Invalid IOI error

Hex Code	Error Message
0027	Unexpected attribute in list
0028	Device Net Error-Invalid member ID
0029	DeviceNet error – member not settable

7.1.1 Determining the Instrument Being Serviced by the AIM

Due to the nature of the Q.iMPACT Matroller, only one instrument on a Bridge may be sent a command at a given time. When you diagnose a particular problem it will be beneficial for you to determine which instrument's command is currently being processed by the AIM.

To determine this, view the value in the Bridge data structure, Assembly slot register (AIMBridge[ ].Internal.AssemblySlot) for the Bridge in question.

7.1.2 Current and Previous Hour Diagnostic Counts

The AIM contains several diagnostic counts designed to report the status of each configured Bridge. These diagnostic counts are reported for the current hour, and the previous hour. The diagnostic counts are not a running average, but a cumulative count which is reset every hour. At the end of an hour, the current count is moved to the previous count and the current count is zeroed.

Monitor these counts over time to determine the overall health of communications between the Controller and the Q.iMPACT Matroller.

7.1.3 AIMBridge[ ].Response.CommandStatus: Diagnostics

In addition to communications diagnostics, the CommandStatus element of the Response Data_ returned from the Q.iMPACT Matroller may be used to diagnose problems.

Status Class and Status Type define the Command Status returned from the Q.iMPACT Matroller, as categorized by the AIM.

There are three Status Classes:

1. Success
2. Warning
3. Error

Within the ERROR Status Class there are four Status Types.

1. Bad Data
2. Communications
3. Instrument
4. Process Error

The following table summarizes the Error Status Class responses.

Table 49: AIMBridge[ ].Response.CommandStatus – Error Class

Value	Status Class	Description	Status Type
7	ERROR	Communications Error	Bad Data
8	ERROR	Invalid Instrument Number	Bad Data
9	ERROR	Invalid Command	Bad Data
10	ERROR	Invalid Material-Path Table Index Number	Bad Data
11	ERROR	Invalid Algorithm in Material-Path Table Entry	Bad Data
12	ERROR	Invalid Feed Type in Material-Path Table Entry	Bad Data
13	ERROR	Invalid Measuring Device Instrument Table Index in Material Path Table Entry	Bad Data
14	ERROR	Invalid Gain In Weight Feed and Dump to Empty Algorithm Combination in Material Path Table	Bad Data
15	ERROR	Invalid Destination in Material Path Table Entry.	Bad Data
16	ERROR	Other invalid data in Material Path Table Entry	Bad Data
17	ERROR	Overlap Feed Request Error, including invalid Loss In Weight Feed in Material Path Entry and Overlapping Feed Command.	Bad Data
18	ERROR	Invalid data In Measuring Device Instrument Table Entry	Bad Data
19	ERROR	Invalid Mode for Command, e.g., PLC is requesting to start a new material transfer before the last feed is complete or before the PLC has acknowledged that the last material transfer is complete.	Process Error
20	ERROR	Requested add amount too small	Process Error
21	ERROR	Requested add amount would bring Scale Device over capacity	Process Error
22	ERROR	Scale Device Currently over Capacity	Process Error
23	ERROR	Scale Device Currently under Zero.	Process Error
24	ERROR	Instrument Malfunction	Instrument
25	ERROR	Target Weight is less than Spill	Process Error
26	ERROR	Response Timeout	Instrument
27	ERROR	Too many overlapping feeds	
30	ERROR	Invalid overlap group number	Bad Data
33	ERROR	Not Enough Material	Process Error
34	ERROR	Flow Meter Configuration Error	Bad Data
100	ERROR	Command Failed	Communication
101	ERROR	Material Transfer Ended Without Request for Acknowledge	Communication

7.1.3.1 Bad Data

This Status Type indicates that the data contained within the command issued to the Q.iMPACT Matroller is invalid.

The User Application and Q.iMPACT Matroller configuration should be verified.

7.1.3.2 Communications

The AIM generates this error based on a failure to send a command and receive the appropriate Command Read within the configured period of time, for the configured number of attempts.

7.1.3.3 Instrument

This Status Type indicates that there is a problem with the Q.iMPACT Matroller, and/or the instrument being accessed.

7.1.3.4 Process Error

If an attempt to execute a feed is made, which is beyond the bounds of the current process conditions, the resulting Command Status will be of this Status Type.

8 Programming Considerations

There are several programming items which must be considered when interfacing to the AIM. This section explores some of these considerations in greater detail.

8.1 Important Notes

Note the following when applying the AIM:

1. During a particular material feed, if the system power is lost, the Q.iMPACT system will terminate all active material feeds. The status reported back in the cyclic message will reflect this condition to the PLC. The AIM will not contain logic to attempt to recover this condition but will update the appropriate status. Recovery logic will be the responsibility of the equipment module.
2. This design assumes that each assembly slot within a Q.iMPACT cluster is assigned a unique instrument number. There is no error detection within the AIM to account for duplicate instrument numbers.
3. The PLC is updated at the ControlNet frequency. Data in the general assembly point is updated at once a second. The critical events are updated on event so that the PLC can move on at the maximum rate.

8.2 Sending a Command

8.2.1 CDReady

Use the following logic to condition the rung which writes command data and sets the AIMInstrument[ ].Handshake.CDReady flag. In the example, the UserConditions bit represents whatever user condition is used to initiate the command to the AIM, and the UserDefinedOneShot is optional depending on the conditions used to trigger the command.

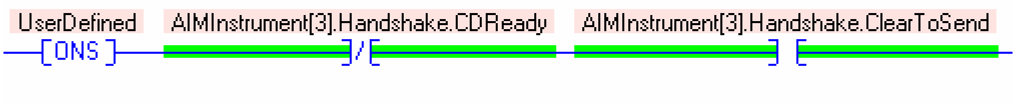

UserDefined
ONS
AIMInstrument[3].Handshake.CDReady
/
AIMInstrument[3].Handshake.ClearToSend

Figure 31: CDReady Conditioning

The following logic shows the writing of command data for transfer to the Q.iMPACT Matroller, and setting of AIMInstrument[ ].Handshake.CDReady flag.

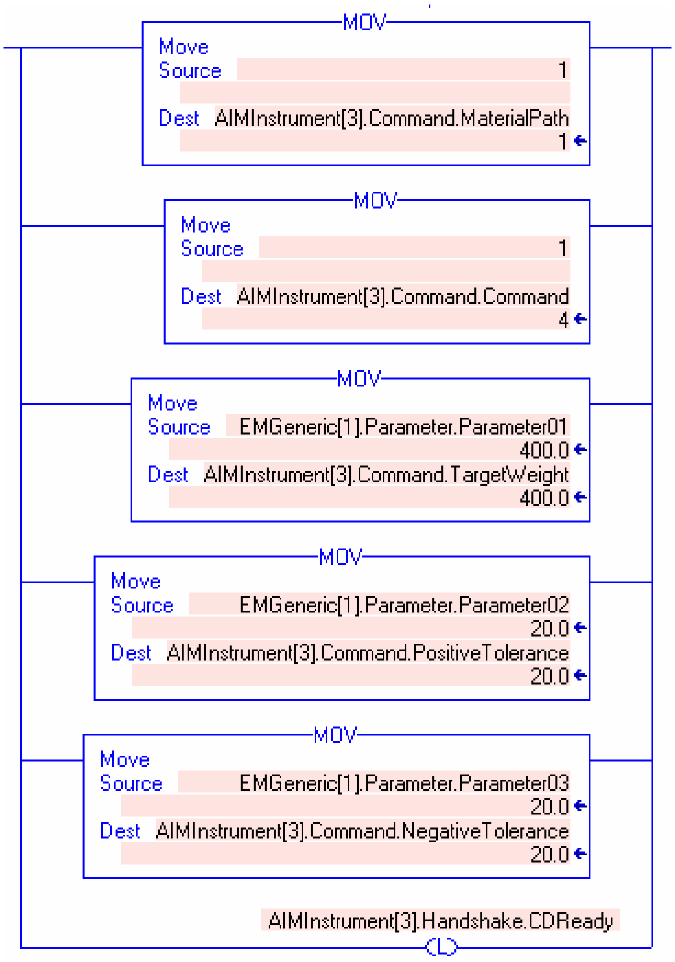
In this example:

- The Instrument is at index 3 (AIMInstrument[3])
  The Material Path is 1, this must be setup within the Q.iMPACT Matroller, this value is written to AIMInstrument[3].Command.MaterialPath

- The Command is 1, Start Material Transfer, this value is written to AIMInstrument[3].Command.Command.

- The Target Weight is 400, Units are defined within the Q.iMPACT Matroller, and this value is written to Instrument [3] .Command.TargetWeight.

- The Positive and Negative tolerances are 20, Units are the same as Target Weight, this value is written to AIMInstrument[3].Command.PositiveTolerance and AIMInstrument[3].Command.NegativeTolerance.



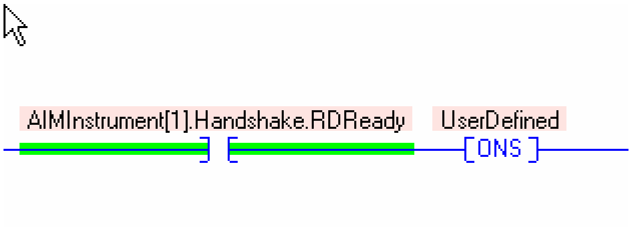
```
MOV
Move
Source                                        1

Dest   AIMInstrument[3].Command.MaterialPath
                                              1

MOV
Move
Source                                        1

Dest   AIMInstrument[3].Command.Command
                                              4

MOV
Move
Source   EMGeneric[1].Parameter.Parameter01
                                          400.0
Dest   AIMInstrument[3].Command.TargetWeight
                                          400.0

MOV
Move
Source   EMGeneric[1].Parameter.Parameter02
                                           20.0
Dest   AIMInstrument[3].Command.PositiveTolerance
                                           20.0

MOV
Move
Source   EMGeneric[1].Parameter.Parameter03
                                           20.0
Dest   AIMInstrument[3].Command.NegativeTolerance
                                           20.0

AIMInstrument[3].Handshake.CDReady
              <L>
```

Figure 32: Writing Command Data to the AIM

8.2.2 RDReady

Using the RDReady flag within an application is optional. The RDReady flag is set (transition high) when a response to a given command is received from the Q.i. Matroller within the AIM. For instance if a given application needed to make a decision based on Command Status, the RDReady flag would be used as a trigger within the application to indicate that the Command Status for the last command is available within the Instument[ ].Response.CommandStatus.

The following figure details the appropriate RDReady conditioning logic.



AIMInstrument[1].Handshake.RDReady

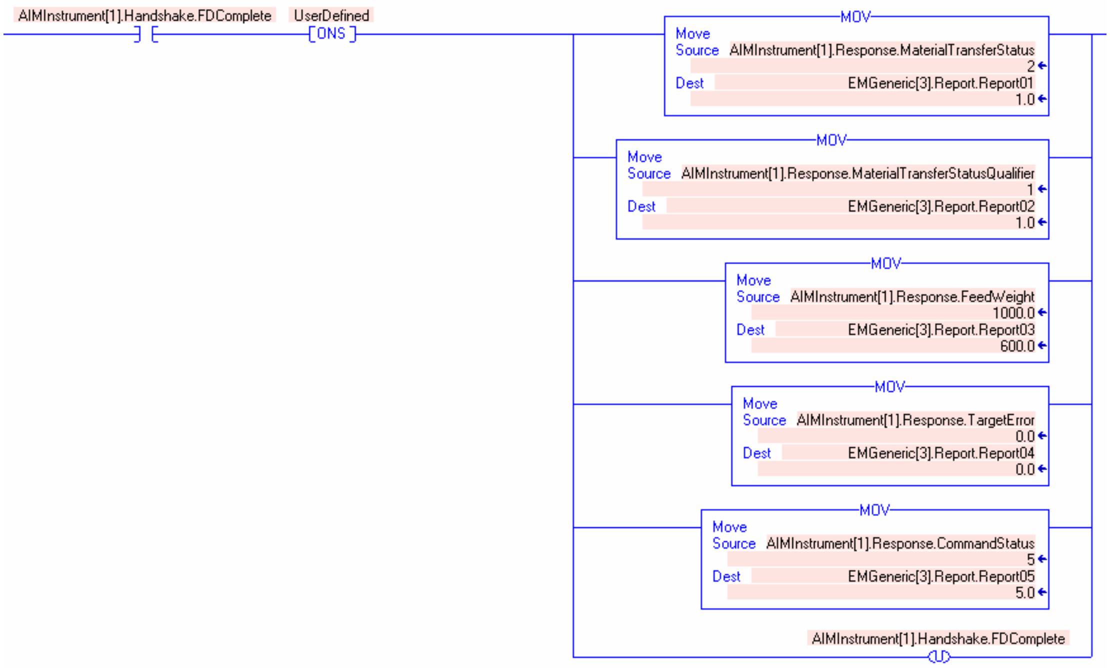
UserDefined

ONS

Figure 33: RDReady Conditioning

8.3 Feed Complete Handshaking

Use the following logic to condition the rung which transfers end of feed data and resets the AIMInstrument[ ].Handshake.FDComplete flag. The UserDefinedOneShot is optional.



AIMInstrument[1].Handshake.FDComplete

UserDefined

ONS

MOV
Move
Source AIMInstrument[1].Response.MaterialTransferStatus
2
Dest EMGeneric[3].Report.Report01
1.0

MOV
Move
Source AIMInstrument[1].Response.MaterialTransferStatusQualifier
1
Dest EMGeneric[3].Report.Report02
1.0

MOV
Move
Source AIMInstrument[1].Response.FeedWeight
1000.0
Dest EMGeneric[3].Report.Report03
600.0

MOV
Move
Source AIMInstrument[1].Response.TargetError
0.0
Dest EMGeneric[3].Report.Report04
0.0

MOV
Move
Source AIMInstrument[1].Response.CommandStatus
5
Dest EMGeneric[3].Report.Report05
5.0

AIMInstrument[1].Handshake.FDComplete

U

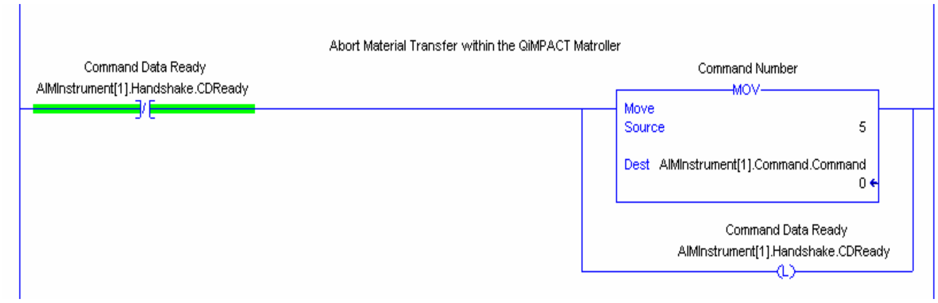
Figure 34: FDComplete Reset Conditioning

The example also shows the transfer of command response data from the AIM, and resetting of AIMInstrument[ ].Handshake.FDComplete flag.
In this example:

- The Instrument is at index 1 (AIMInstrument[1])
- The command response data is transferred to the EM’s report data registers.

8.4 Abort Handshaking

Use the following logic to perform an abort on a material transfer running within the Q.i. Matroller. Note that this same logic (using different command numbers) can be used for instrument and cluster resets as well.



Abort Material Transfer within the QiMPACT Matroller

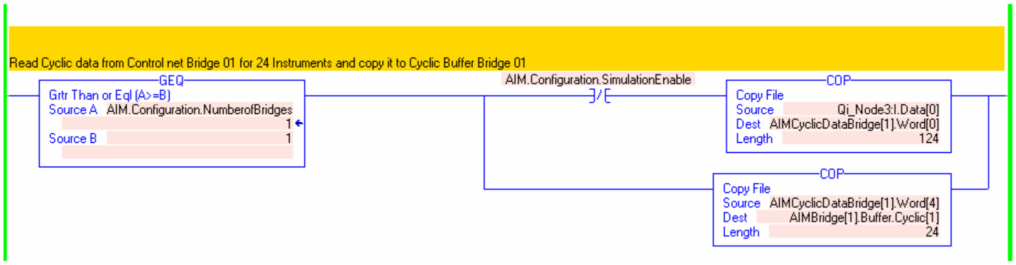
Command Data Ready
AIMInstrument[1].Handshake.CDReady

Command Number
MOV
Move
Source 5
Dest AIMInstrument[1].Command.Command
0

Command Data Ready
AIMInstrument[1].Handshake.CDReady
L

8.5 ControlNet Configuration Programming

For each of the ten possible Bridges which may be defined within the AIM, a line of code exists within the AIMCyclicRead program file. The rung contains a COP (copy) instruction which copies the 124 double words of data from the data space defined within the ControlNet configuration, into the AIM’s internal data structure. The user must edit the ‘source’ within the copy instruction, for each desired bridge.



Read Cyclic data from Control net Bridge 01 for 24 Instruments and copy it to Cyclic Buffer Bridge 01

GEQ
Grtr Than or Eql (A>=B)
Source A AIM.Configuration.NumberofBridges
1
Source B 1

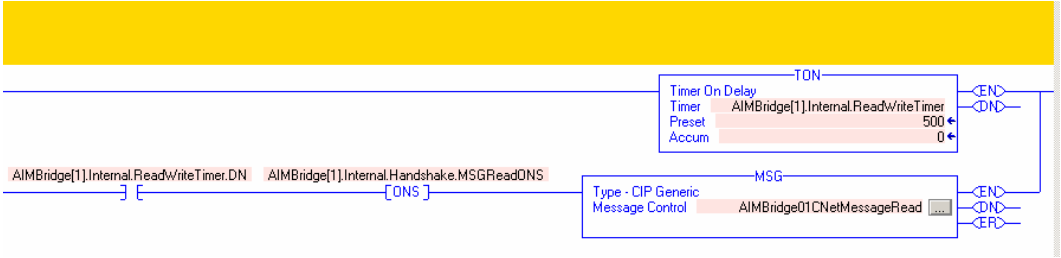
AIM.Configuration.SimulationEnable

COP
Copy File
Source Qi_Node3:I.Data[0]
Dest AIMCyclicDataBridge[1].Word[0]
Length 124

COP
Copy File
Source AIMCyclicDataBridge[1].Word[4]
Dest AIMBridge[1].Buffer.Cyclic[1]
Length 24

Figure 35: Cyclic Data, ControlNet Programing

For each of the ten possible Bridges which may be defined within the AIM, a line of code exists within the AIMMSGRead program file. The rung contains a MSG (message) instruction, which provides the explicit ControlNet read functionality. The user must edit this MSG instruction, and configure the correct CNet Node for the Bridge. The user must edit the MSG instruction, for each desired bridge. See the “ControlNet” section of this document for all ControlNet configuration details.


TON
EN
DN
AIMBridge[1].Internal.ReadWriteTimer.DN
AIMBridge[1].Internal.Handshake.MSGReadONS
ONS
MSG
EN
DN
ER

Timer On Delay
Timer AIMBridge[1].Internal.ReadWriteTimer
Preset 500
Accum 0

Type - CIP Generic
Message Control AIMBridge01CNetMessageRead

Figure 36: Explicit Message Read, ControlNet Programming

For each of the ten possible Bridges which may be defined within the AIM, a line of code exists within the AIMMSGWrite program file. The rung contains a MSG (message) instruction, which provides the explicit ControlNet write functionality. The user must edit this MSG instruction, and configure the correct CNet Node for the Bridge. The user must edit the MSG instruction, for each desired bridge. See the “ControlNet” section of this document for all ControlNet configuration details.

8.6 Monitoring Error Conditions

The AIM’s error handling depends on the user application. This section deals with the items that must be considered when writing logic to monitor error conditions.

8.6.1 CommandNull and MaterialPathNull

If a null (zero) Command or Material Path is sent to the Instrument Command data, the following occurs:

- AIM will not send command to the Q.iMPACT Matroller.
- AIM will not set FDComplete.
- AIM will set AIMInstrument[ ].Handshake.CommandNull and AIMInstrument[ ].Handshake.MaterialPathNull.
- AIMInstrument[ ].Handshake.CDReady will be unlatched (turned off) by the AIM.

8.6.2 Command Status

The user application should monitor for errors, as indicated with Status Class and Status Type flags within the AIMInstrument[ ].Handshake data, and the command status in the AIMInstrument[ ].Response.CommandStatus register.

8.6.3 Material Transfer Status

The user application should monitor for errors, within the AIMInstrument[ ].Response.MaterialTransferStatus register.

8.6.4 Material Transfer Status Qualifier

The user application should monitor for overweight and underweight status qualifiers, within the AIMInstrument[ ].Response.MaterialTransferStatusQualifier register.

Siempelkamp Canada Inc.
400 Collier-Macmillan Drive
Cambridge, Ontario N1R 7H7
Canada

P/N: 10AIM-DOCIMPC
(03/05) 1.2

Printed in Canada