iVu Plus Communications

Instruction Manual

Original Instructions
B_3084218 Rev. D
18 March 2014

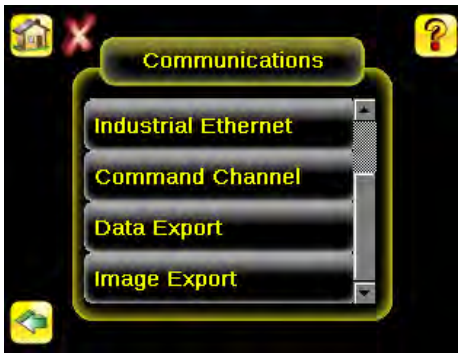
Contents

1 iVu Plus Communication Summary of Ethernet and Serial

The iVu Plus communicates with other devices via Ethernet or a UART serial communications port (RS-232). In order to establish an Ethernet connection to the sensor, the external device must be configured with the correct IP address and TCP port to communicate. To use the serial communications connection, port settings for baud rate, data bits, parity, and stop bits must be configured on the iVu Plus to match the settings of the external device.

1.1 Communication Channels

The iVu Plus TG supports up to four communications channels. To access the channels, go to Main Menu > System > Communications.



- Command Channel—a bi-directional communication protocol that currently supports ASCII and enables other devices to remotely control the iVu Plus sensor and access sensor results
- Industrial Ethernet—a bi-directional communication channel that allows the user to control the sensor and access sensor results using Ethernet/IP, Modbus/TCP, or PCCC protocol
- Data Export—used to export selected inspection data to a remote device
- Image Export—used to export inspection images to a remote device

Data export and command channel can be configured for either Ethernet or Serial I/O (but not both); image export is only available over Ethernet. The table below briefly summarizes valid communication channel configuration options.

Command Channels	Scenario #1		Scenario #2		Scenario #3	
	Ethernet	Serial I/O	Ethernet	Serial I/O	Ethernet	Serial I/O
Command Channel	Yes	No	No	Yes	Yes	No
Industrial Ethernet	Yes	No	Yes	No	Yes	No
Data Export	Yes	No	Yes	No	No	Yes
Image Export	Yes	No	Yes	No	Yes	No

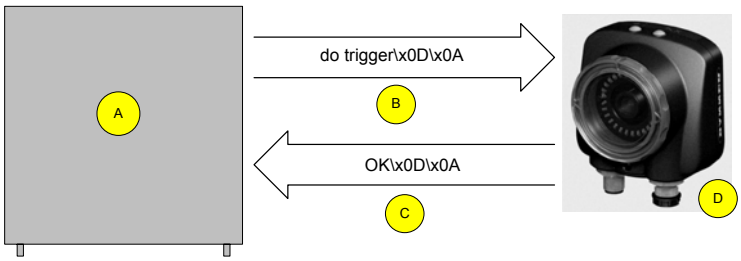
1.2 Industrial Ethernet

Main Menu > System > Communications > Industrial Ethernet

The iVuPlus device can be controlled or monitored over Industrial Ethernet using Ethernet/IP, Modbus/TCP or PCCC protocols. This document will help you to set up the iVu Plus in the desired configuration and provide you with information you will need to connect to the master device (PLC, HMI, etc.).

1.3 Command Channel

The iVu Plus TG command channel is a bi-directional communication protocol that currently supports ASCII via either Ethernet or the RS-232 serial interface, and enables other devices to remotely control the iVu sensor and access sensor results.



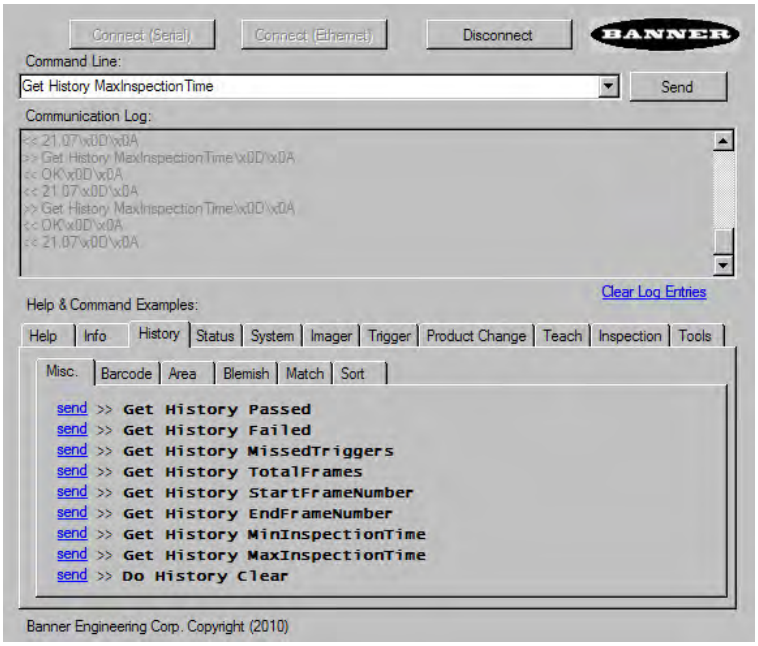
A	Control Device, which can be a PLC, PC program, or a terminal	C	Response Frame
B	Request Frame	D	iVu Plus TG Sensor

The following are some of the functionality available via the command channel:

- Get sensor information (such as version and sensor name)
- Control "discrete" I/O (such as trigger and teach)
- Get sensor results (such as sensor status)
- Change the running inspection

Command Channel Sample Application

The iVu Plus TG installation CD has a Command Channel sample application that provides an easy interface to execute commands. In a production environment, you will need to create your own application for bi-directional communication with the sensor.



1.4 Data Export

The iVu Plus sensor provides for exporting user-selected inspection data via either Ethernet or the RS-232 serial interface. Inspection data that can be exported includes:

- Pass/Fail Output
- Inspection Name
- Sensor Result
    - Name

- Pass/Fail
- Sensor Result (see *Table 1* on page 5 for additional information)

- Inspection Time (ms)

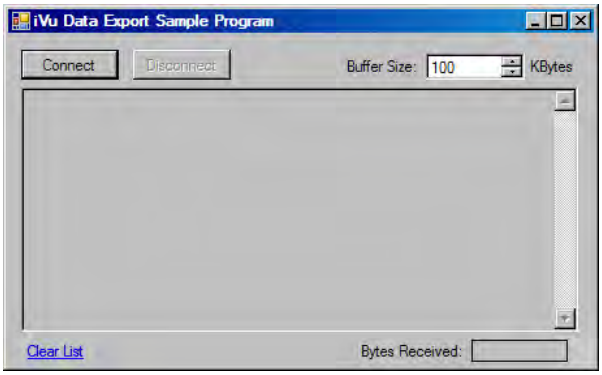
Table 1: Sensor Results

Sensor Type	Data to Export
Area	Count
	Area Range
Blemish	Count
	Edge Length Range
Match	Count
	Percent Match
Sort	Count
	Percent Match
	All Found Pattern Numbers
	All Found Pattern Names

Data export settings apply to all inspections sensor-wide. If items are selected that are not part of the current inspection, those items are ignored.

Sample Application

The iVu Plus TG installation CD has a Data Export sample application that provides for viewing exported data while setting up the sensor, etc. In a production environment, you will need to create your own application to process data exported from the sensor.



1.5 Image Export

Image export is only supported on Ethernet ports. Inspection images are a maximum 320×240 8-bits per pixel grayscale images in Windows BMP format that represent all the data in a full Field of View (FOV).

Each exported image is comprised of a header (64 bytes) followed by the image data (approximately 78K). All 16- and 32-bit numeric entries are little endian.

The header includes the following information:

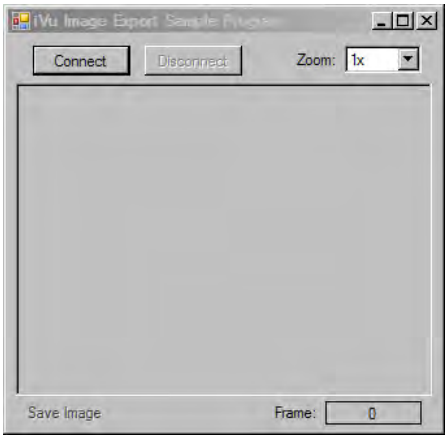
Byte Offset	Field	Size in Bytes	Data Type	Description
0-15	Header Prefix	16	char	"IVU PLUS IMAGE"
16-19	Header Version	4	UInt32	1
20-23	Image Size	4	UInt32	Number of bytes (Windows BMP image)

Byte Offset	Field	Size in Bytes	Data Type	Description
24-27	Image Frame Number	4	UInt32	Most recently snapped image frame number
28-29	Image Width	2	UInt16	320 (max)
30-31	Image Height	2	UInt16	240 (max)
32-33	Image Format	2	UInt16	0: Bitmap, 1: JPEG
34-63	Reserved	32	byte	Reserved for future use

NOTE: If FOV's are adjusted so that they are smaller, the bitmaps will also be smaller.

Image Export Sample Application

The iVu Plus TG installation CD has a Image Export sample application that provides a way to save exported images. In a production environment, you will need to write your own application to process exported images, for example to display them on an HMI or to save them to disk.



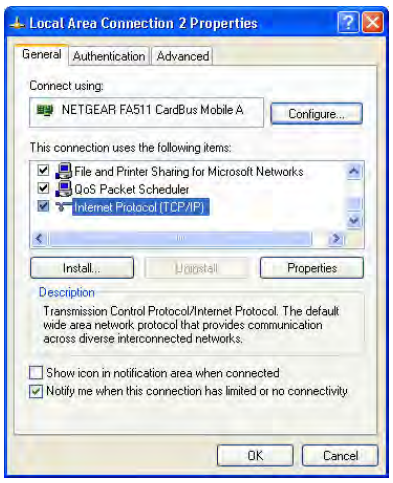
2 Enabling Communications

2.1 Setting Up Ethernet Communications

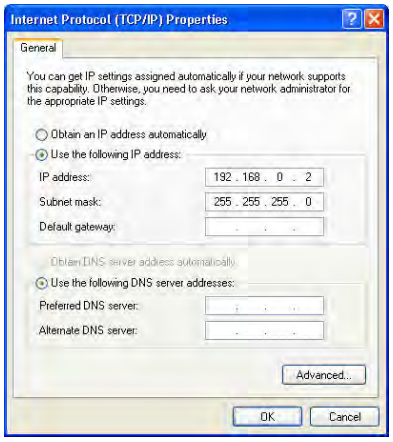
Configure both the PC and the sensor using the following instructions.

2.1.1 Windows XP

1. Open Network Properties on the PC (right-click on the Network Neighborhood icon).
2. On the Local Area Connection, right-click on Properties.
3. In the dialog, click on Internet Protocol (TCP/IP) and click the Properties button.

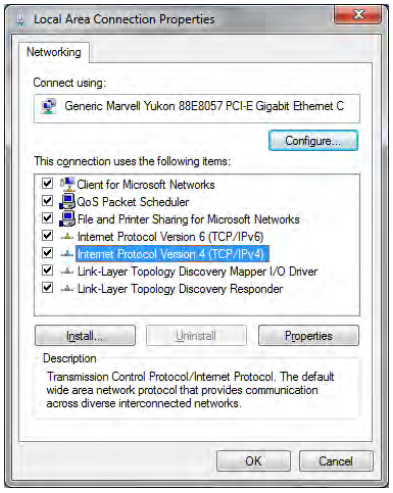


4. In the Internet Protocol (TCP/IP) Properties dialog, select Use the following IP address and make sure that the IP address is 192.168.0.2, and the subnet mask is 255.255.255.0.

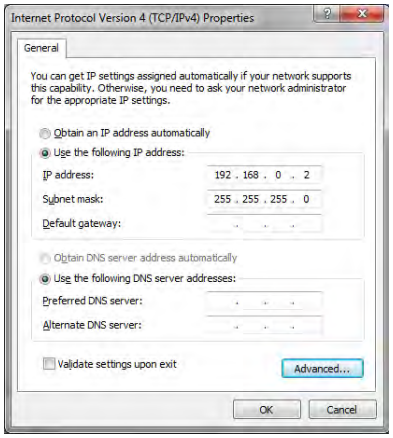


2.1.2 Windows 7

1. Open Network Connections by clicking on the Start button, then selecting the Control Panel followed by Network and Internet, and clicking Manage network connections.
2. Right-click the connection you want to change, then click Properties. If you are prompted for an administrator password or confirmation, type the password or provide confirmation.
3. In the Networking dialog, click on Internet Protocol Version 4(TCP/IPv4) and click the Properties button.

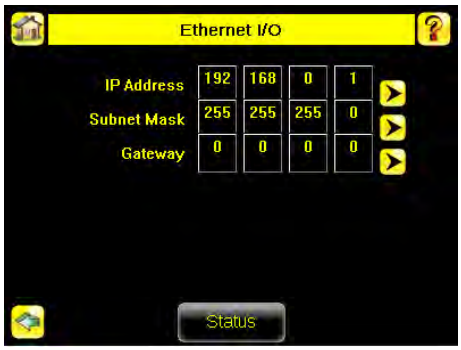


4. In the Internet Protocol (TCP/IPv4) Properties dialog, select Use the following IP address and make sure that the IP address is 192.168.0.2, and the subnet mask is 255.255.255.0.

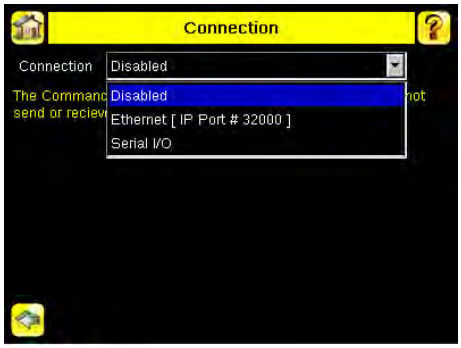


2.1.3 Sensor Setup for Ethernet Communications

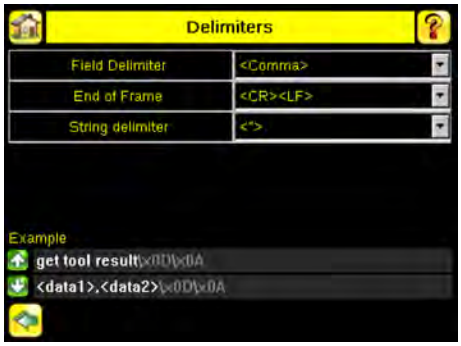
1. Go to Main Menu > System > Communications > Ethernet I/O and make sure that the sensor is configured as shown below.



2. To enable the command channel over Ethernet:
    a. Go to Main Menu > System > Communications > Command Channel > Connection, and select Ethernet [IP Port # 32000].



b. Configure the field and end-of-frame delimiters. Go to Main Menu > System > Communications > Command Channel > Delimiters.

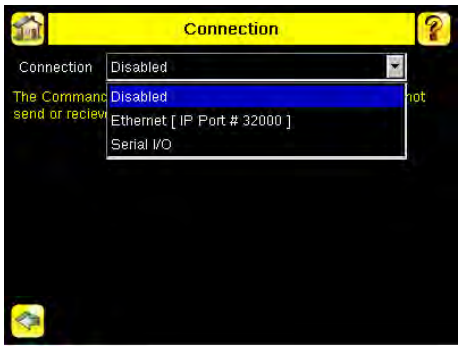


Valid end-of-frame delimiters are: <comma>, <colon>, <semicolon>, <CR>, <CR><LF>, <LF><CR>, or <ETX>.

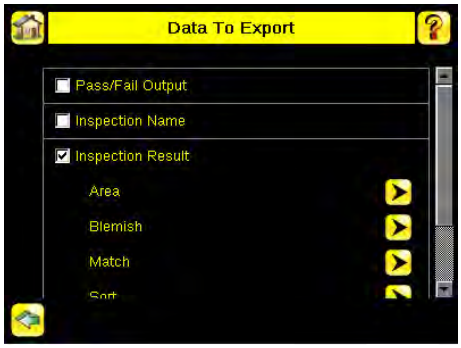
c. Verify that the iVu receives and transmits data correctly.

3. To enable Data Export over Ethernet:

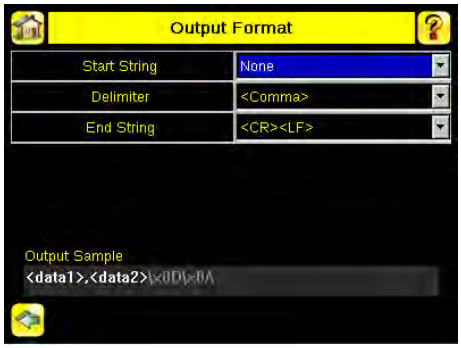
a. Go to Main Menu > System > Communications > Data Export > Connection and select Serial I/O from the drop-down.



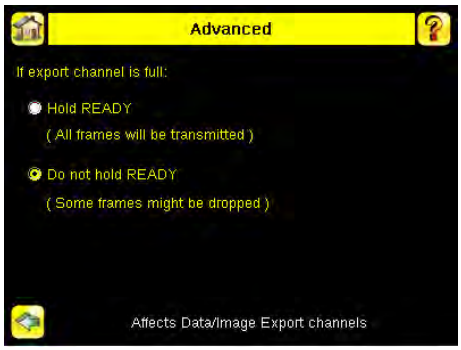
b. Go to Main Menu > System > Communications > Data Export > Data To Export and select the inspection data to export.



c. Go to Main Menu > System > Communications > Data Export > Output Format and select the Start String, Delimiter, and End String.



d. Go to Main Menu > System > Communications > Data Export > Advanced.



During the Data and Image export operation the sensor's output channels might become full. This can occur if the sensor is producing export data (frames) faster than the data can be exported from the device (due to bandwidth limitations) or faster than the client is reading the channel export data.

This setting affects how the sensor will behave in this situation.

- Select Hold READY to ensure that all frames are transmitted. In this case, the READY signal will remain inactive (sensor is busy) until the new frame has been added to the channel for transmission. Triggers might be missed during this time.
- Select Do not hold READY to cause the sensor to discard the new frame if the channel is full and thus activate the READY signal immediately after the current inspection is complete. In this case, the discarded frames will not be transmitted.

2.1.4 Communications Channel Ports

The following are the default Ethernet port settings for the communications channels:

- Command Channel — 32200
- Data Export — 32100
- Image Export — 32000

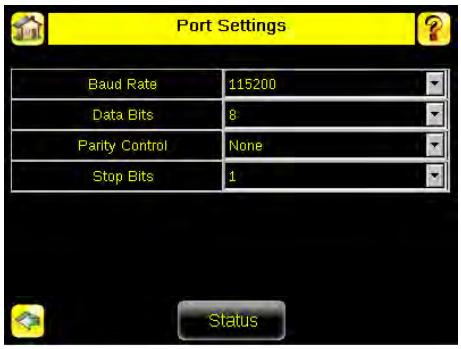
2.2 Setting Up Serial Communications

1. Electrically connect the control device and the iVu sensor. On the iVu, the pins/wire colors used for serial communications via RS-232 are shown in the table below.

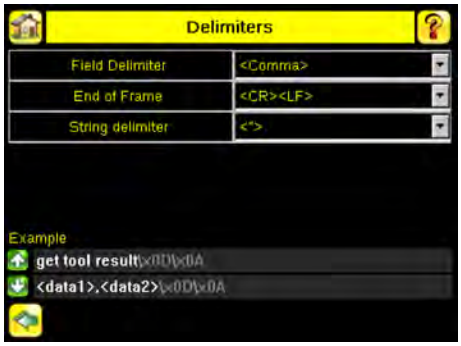
   Table 2: iVu RS-232 Connections

Pin #	Wire Color	Description
10	Light-Blue	TX
11	Black	Signal Ground
12	Violet	RX

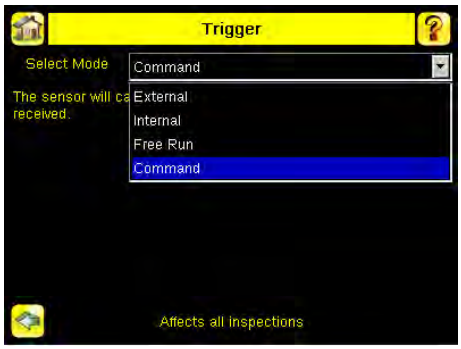
2. Configure port settings (baud rate, data bits, parity, and stop bits) on the iVu to match the settings on the control device. Go to Main Menu > System > Communications > Serial I/O.



3. To enable the command channel over the serial connection:
   a. Go to Main Menu > System > Communications > Command Channel > Connection and select Serial I/O.
   b. Configure the field and end-of-frame delimiters. Go to Main Menu > System > Communications > Command Channel > Delimiters.



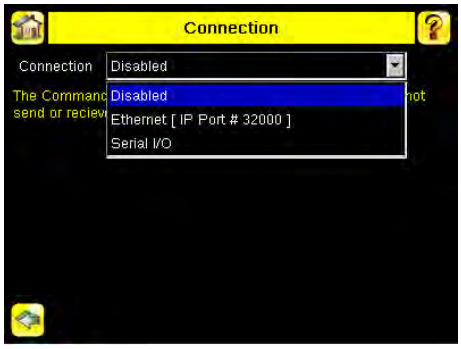
   Valid end-of-frame delimiters are: <comma>, <colon>, <semicolon>, <CR>, <CR><LF>, <LF><CR>, or <ETX>.
   c. Optionally, if you want to trigger the iVu from the control device, set the trigger mode to Command (go to Main Menu > Imager > Trigger and select Command from the drop-down).



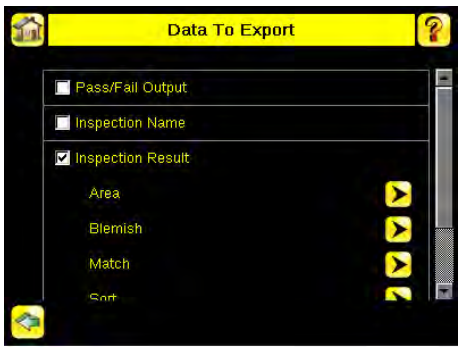
d. Verify that the iVu receives and transmits data correctly.

4. To enable Data Export over the serial connection:

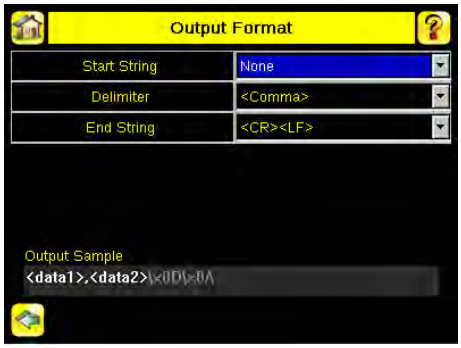
a. Go to Main Menu > System > Communications > Data Export > Connection and select Serial I/O from the drop-down.



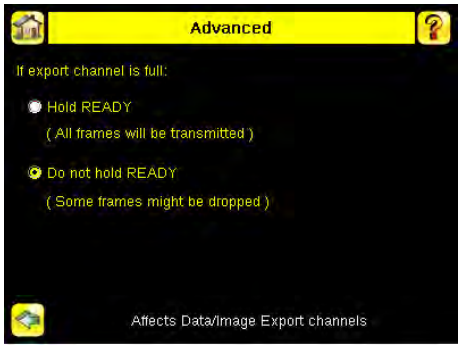
b. Go to Main Menu > System > Communications > Data Export > Data To Export and select the inspection data to export.



c. Go to Main Menu > System > Communications > Data Export > Output Format and select the Start String, Delimiter, and End String.



d. Go to Main Menu > System > Communications > Data Export > Advanced.



During the Data and Image export operation the sensor's output channels might become full. This can occur if the sensor is producing export data (frames) faster than the data can be exported from the device (due to bandwidth limitations) or faster than the client is reading the channel export data.

This setting affects how the sensor will behave in this situation.

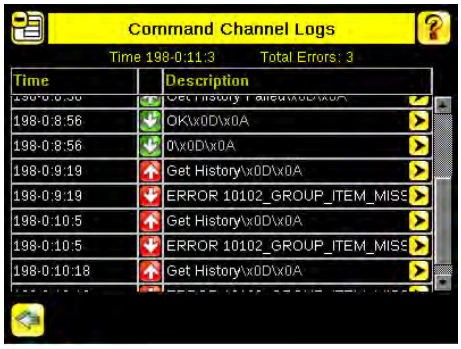
- Select Hold READY to ensure that all frames are transmitted. In this case, the READY signal will remain inactive (sensor is busy) until the new frame has been added to the channel for transmission. Triggers might be missed during this time.
- Select Do not hold READY to cause the sensor to discard the new frame if the channel is full and thus activate the READY signal immediately after the current inspection is complete. In this case, the discarded frames will not be transmitted.

3 Testing and Troubleshooting iVu Plus Communications

3.1 Understanding the Communication Log

The iVu Plus sensor includes the following Communication Logs:

- Command Channel Log that can be used to ensure that commands are properly formed (syntax is correct), and provides a history of commands issued along with responses to these commands. To access the Command Channel Log, go to Main Menu > Logs > Communication Logs > Command Channel
- Data Export and Image Output logs that are purely output logs (that is, there is no receive activity to log)



Some notes about the logs:

- To see an expanded view of each entry, click on the small right-triangle control on each entry
- To save the log, click the save icon. The saved communication log can be loaded into the emulator for troubleshooting offline

The table below describes the icons used in the Communication Log, the up-arrow indicates an incoming request to the iVu from the control device; the down-arrow indicates an outgoing response from the iVu to the control device.

Icon	Description
	Port opened.
	Port closed.
	Indicates that the command has been processed without errors.
	Indicates that the incoming entry is stalled (no new bytes), or end-of-frame delimiter was not received, or client is not reading data on ethernet.
	If the response frame contains an error or is dropped, the log entry icons for the request and the response frames will be colored red, and the displayed error count will increment by one.
	If the command takes a long time to process, the last long entry will change to an hourglass (for example, during trigger of long inspections).

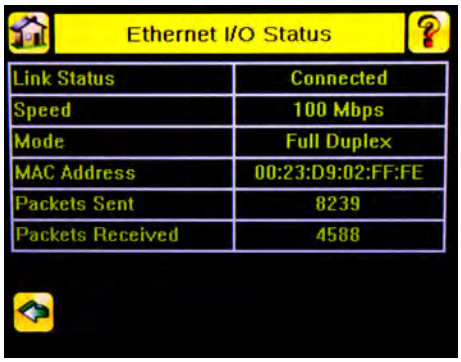
For Ethernet channels:

- The channel's log will show an Enabled entry that indicates which port is being listened to
- When a client connects, a log entry is added that indicates which IP address connected
- When a client closes the connection, a log entry indicates that the channel is no longer being listened to

3.2 Ethernet I/O

3.2.1 Ethernet I/O Status

The Ethernet I/O Status screen can be used to verify that the Ethernet wiring has been correctly set up. In addition to determining if the link has been established, incoming and outgoing traffic can be monitored.



3.3 Serial I/O

3.3.1 Using the Port Status Screen for Testing RS-232 Communications

The Port Status screen can be used to ensure data is entering and exiting the sensor. This can be useful for debugging issues such as improper wiring, mismatched baud rates, or other serial I/O issues. To access the Port Status screen, go to Main Menu > System > Communications > Serial I/O and click on the Status button.

- The upper field shows the bytes received (request frame) on the iVu from the control device.
- The lower field shows the bytes sent (response frame) from the iVu to the control device.

Port Errors

The Port Errors screen can help to debug communications channel issues: Parity, Break, and Framing indicate mismatched port settings or, in the case of Break, incorrect cabling.

3.3.2 Using the iVu Command Channel Sample Application or a Terminal Program for Testing

The easiest way to test that the iVu command channel is correctly receiving and transmitting data is to use either the iVu Command Channel Sample App (available on the installation CD) or to use a terminal program running on a PC:

If using a terminal program, in the terminal program's configuration:

- Set new-line transmit to <CR><LF> (and set the end-of-frame delimiters on the iVu to match).
- Enable local echo.
- Set the Serial port set up so that the PC port number's baud rate, data, parity, and stop bits match those setup on the iVu.

3.3.3 Verifying Basic Receive Functionality

To verify the iVu can receive request frames from the requesting device:

1. On the iVu Sensor, go to the Main Menu > System > Communications > Serial I/O > Port Status screen.

2. On the requesting device, transmit one or more bytes to the iVu sensor.
    - If the data byte values appear correct and the number sent by the requesting device matches the number received by the iVu sensor, then the transmit/receive functionality is working properly.
    - If the connection is incorrect (electrically) or if the baud rate is mismatched, no bytes will appear in the upper field on the Port Status screen.
    - If the connection is correct (electrically), bytes will appear in the upper field of the Port Status screen in the order they were received.
    - If the Port Status: Errors at the bottom of the Port Status screen highlights red, then the connection is correct electrically but there is likely a settings mismatch between the iVu sensor and the requesting device. Verify the settings on both devices.
    - If the bytes appear with no errors but appear incorrect or there are too many or too few, then the port settings (for example, baud rate) are likely mismatched in a way that does not generate serial hardware errors. Verify the settings on both devices match exactly.

3.3.4 Verifying Basic Transmit Functionality

The iVu command channel will only send response frames to the requesting device if it receives a valid end-of-frame delimiter from the requesting device. To verify transmit functionality:

1. Transmit an end-of-frame delimiter sequence from the requesting device to the iVu sensor. For example, in a terminal program, simply hit Enter.

    If a valid end-of-frame delimiter is received, the iVu sensor will immediately transmit a short error message back to the requesting device (for example, ERROR 10000_COMMAND_MISSING).
2. Verify that the number of bytes sent by the requesting device are the same as the number shown in the lower field of the Port Status screen on the iVu sensor. Go to the Main Menu > System > Communications > Serial I/O > Port Status screen.

3. If the byte count does not match, re-verify that the settings on both devices match exactly. If no bytes are received, re-check the wiring.

If the correct response frame is received, then basic electrical and port settings are correct.

4 Command Channel Primer

4.1 Command Channel Commands

All iVu command channel request command frames use the following syntax:

```
>> command group item value<EOF>
```

Notes
<EOF> is the end-of-frame delimiter. See below for a description.
All commands are in ASCII and are case-insensitive

command
An action to be performed on a particular iVu group; for example, get, set, do, login, or logout.

group
Identifies the iVu group that the command should act upon; for example, info, system, trigger, or bcr_input.

item
Further qualifies the action by specifying an item within the identified group; for example, comparedata or status.

value
For set commands, this identifies the data that must be set for the specified group item.

Note: Item is not used with get commands.

<EOF>
Identifies the end-of-frame for the command so that the iVu knows to begin processing. The iVu will return a response that includes the end-of-frame delimiter. The options for the <EOF> are set in the iVu Serial I/O menu, and are as follows:

- <comma>
- <colon>
- <semicolon>
- <CR>
- <CR><LF>
- <LF><CR>
- <ETX>

NOTE: When data is displayed on iVu screens such as the Port Status screen, printable delimiters are displayed as expected. Non-printable characters, such as <CR> are displayed in hex notation (\x0D).

4.1.1 Command Flow

The command flow should be such that a new command request should not be issued until the iVu command channel acknowledges the previous command request.

For example, the following is a series of command requests and responses. The first request sets the trigger mode to command and, once the sensor responds with an "OK," the next command request is issued to do (or execute) the trigger.

```
>>  set trigger mode command\x0D\x0A
<<  OK\x0D\x0A
>>  do trigger\x0D\x0A
<<  OK\x0D\x0A
```

4.1.2 String Delimiters and Escaping

By default setting, all strings used in commands are enclosed in quotation marks (""). All text in quotes is part of the command. Quotes (") or back-slashes (\) that are part of the string must be escapted with a back-slash. For example:

"abc\"def\"ghi\\jkl"

Set the String Delimiter parameters to 'None' if strings should not be enclosed in quotation marks.

4.1.3 Command Channel Command Synopsis

There are a number of general types of commands to do, set, and get sensor data.

Command Channel Response Frames

The iVu responds to all request frames with one or two responses depending on the type of command.

Do commands

All do commands are followed by one response that identifies the command status. For example:

```
>>  do trigger\x0D\x0A
<<  OK\x0D\x0A
```

Get commands

All get commands are followed by two responses: the first identifies the status of the command, and the second contains the retrieved information. For example:

```
>>  get bcr_input comparedata\x0D\x0A
<<  OK\x0D\x0A
<<  "012345ABCDEF"\x0D\x0A
```

Set commands

All set commands are followed by one response that identifies the command status. For example:

```
>>  set bcr_input comparedata "012345ABCDEF"\x0D\x0A
<<  OK\x0D\x0A
```

Command Channel Command Status

The command status is either OK or ERROR. If OK, then the command has fully and successfully completed. If an error is returned it is in the form *ERROR nnnnn_ERROR_IDENTIFIER* (for example ERROR 10001_COMMAND_NOT_RECOGNIZED). Refer to *Command Channel Error Codes* on page 26 for a list of errors.

4.2 Conventions Used for Examples

There are a number of command channel examples included here, and the following are the conventions used in the examples:

- All examples use <CR><LF> for the end-of-frame delimiter, and this delimiter is always denoted in hex (\x0D\x0A) since that is what is displayed in the iVu logs and, for example, the Port Status screen.
- All commands are in bold text.
- For each example, a command request to the iVu sensor is prefaced with a >>, and a command response frame from the iVu sensor is prefaced by a << as shown below. These are only used to make the documentation clearer.

```
>>  get info companyname\x0D\x0A
<<  OK\x0D\x0A
<<  "Banner Engineering Corp."\x0D\x0A
```

4.3 Examples

4.3.1 How to Trigger the Sensor and Retrieve Inspection Data using the Command Channel

This example is based on a sort inspection. To trigger the sensor and retrieve inspection data, do the following

1. Make sure that the Command Channel is enabled using either Ethernet or Serial I/O (Main Menu > System > Communications > Command Channel > Connection).
2. Set Trigger to Command. Go to the Main Menu > Imager > Trigger screen, and from the drop-down select Command.
3. Issue a trigger command as follows:

```
>>  do trigger\x0D\x0A
<<  OK\x0D\x0A
```

4. Check that the inspection passed.

```
>>  get inspection status\x0D\x0A
<<  OK\x0D\x0A
<<  Pass\x0D\x0A
```

5. Get the pattern names that are stored in the iVu sensor.

```
>>  get sort_result patternnames\x0D\x0A
<<  OK\x0D\x0A
<<  "pattern_1","pattern_2"x0D\x0A
```

4.3.2 How to Execute a Product Change Using the Command Channel

1. Make sure that the Command Channel is enabled using either Ethernet or Serial I/O (Main Menu > System > Communications > Command Channel > Connection).
2. Get all the stored inspection names.

```
>>  get productchange inspectionnames\x0D\x0A
<<  OK\x0D\x0A
<< "Inspection 1", "Inspection 2", "Inspection 3"\x0D\x0A
```

3. Execute a product change.

```
>>  do productchange "inspection2"\x0D\x0A
<<  OK\x0D\x0A
```

4. Check that the inspection passed.

```
>>  get inspection status\x0D\x0A
<<  OK\x0D\x0A
<<  Pass\x0D\x0A
```

4.4 Command Channel Reference

4.4.1 Info Command Group

Command	Group	Item	Description
Get	Info	CompanyName	The company name as a string.
Get	Info	ModelNumber	The sensor model number as a string.
Get	Info	FirmwareVersion	The sensor firmware version as a string.
Get	Info	SerialNumber	The sensor serial number as a string.
Get	Info	Name	The sensor name as a string.
Get	Info	BootNumber	The number of sensor bootups.
Get	Info	UpTimer	The elapsed time the sensor has been running in the format hh:mm:ss:msec.
Get	Info	HourCount	The number of hours the sensor has been running.
Get	Info	RemoteConnected	The remote display connected status as a boolean value (true or false).
Get	Info	RemoteModelNumber	The model number of the remote display as a string.
Get	Info	RemoteSerialNumber	The serial number of the remote display as a string.

Examples

```
>>  get info companyname\x0D\x0A
<<  OK\x0D\x0A
<<  "Banner Engineering Corp."\x0D\x0A
```

```
>>  get info bootnumber\x0D\x0A
<<  OK\x0D\x0A
<<  42\x0D\x0A
```

```
>>  get info uptimer\x0D\x0A
<<  OK\x0D\x0A
<<  4:42:42:324\x0D\x0A
```

4.4.2 System Command Group

Command	Group	Item	Description
Do	System	Reboot	Reboots the sensor. Pre-empts other commands except Save.
Do	System	Save	Saves inspection and configuration parameters. Blocks until finished. Should be used sparingly.
Get	Ethernet	IPAddress	Get the current active IP address of the sensor as a string.
Get	Ethernet	SubnetMask	Get the current active subnet mask of the sensor as a string.
Get	Ethernet	Gateway	Get the current active Gateway address of the sensor as a string.
Set	Ethernet	IPAddress	Set IP address of the sensor. A valid IP address must be supplied as a string (for example: 192.168.0.1). A 'Reboot' command from the command channel must follow in order to make the new IP address effective. You may also set new Subnet Mask and Gateway address as required before a 'Reboot' command is sent to the sensor.
Set	Ethernet	SubnetMask	Set new subnet mask. A 'Reboot' command is required to be sent from the command channel in order to make the new mask effective.
Set	Ethernet	Gateway	Set new Gateway IP address. A 'Reboot' command is required to be sent from the command channel to make the new address effective.

Examples

```
>>  do system save\x0D\x0A
<<  OK\x0D\x0A
```

4.4.3 Status Command Group

Command	Group	Item	Description
Get	Status	Ready	Flag indicating whether the system is ready to trigger (true) or busy (false).
Get	Status	SystemError	Flag indicating whether a system error is active (true) or cleared (false).
Do	Status	ClearSystemError	Clears the system error LED and sets the internal flag to false.

Examples

```
>>  get status ready\x0D\x0A
<<  OK\x0D\x0A
<<  True\x0D\x0A
```

```
>>  get status systemerror\x0D\x0A
<<  OK\x0D\x0A
<<  False\x0D\x0A
```

```
>>  do status clearsystemerror\x0D\x0A
<<  OK\x0D\x0A
```

4.4.4 Trigger Command Group

Command	Group	Item	Description
Get	Trigger	Mode	Sets trigger mode to one of the valid trigger modes for the sensor.
Set	Trigger	Mode	Sets trigger mode to one of the valid trigger modes for the sensor.

Command	Group	Item	Description
Do	Trigger		Initiates a single trigger. The sensor does not transmit a response until the sensor has completed the action.

Examples

```
>>  set trigger mode command\x0D\x0A
<<  OK\x0D\x0A
```

```
>>  get trigger mode\x0D\x0A
<<  OK\x0D\x0A
<<  Command\x0D\x0A
```

```
>>  do trigger\x0D\x0A
<<  OK\x0D\x0A
```

4.4.5 Imager Command Group

Command	Group	Item	Description
Get	Imager	Gain	The sensor's value used to electronically brighten all image pixels This value can be modified using the sensor's touchscreen. This remotely modified value is not persisted to the sensors permanent memory. The 'Save' operation is required to persist this value.
Set	Imager	Gain	The sensor's value used to electronically brighten all image pixels This value can be modified using the sensor's touchscreen. This remotely modified value is not persisted to the sensors permanent memory. The 'Save' operation is required to persist this value.
Get	Imager	Exposure	The sensor's value used to control the amount of time the imager is allowed to gather light for the image. This value can be modified using the sensor's touchscreen. This remotely modified value is not persisted to the sensors permanent memory. The 'Save' operation is required to persist this value.
Set	Imager	Exposure	The sensor's value used to control the amount of time the imager is allowed to gather light for the image. This value can be modified using the sensor's touchscreen. This remotely modified value is not persisted to the sensors permanent memory. The 'Save' operation is required to persist this value.

Examples

```
>>  get imager exposure\x0D\x0A
<<  OK\x0D\x0A
<<  Command\x0D\x0A
```

```
>>  set imager exposure"11900"\x0D\x0A
<<  OK\x0D\x0A
```

4.4.6 Teach Command Group

Command	Group	Item	Description
Do	Teach	NextTrigger	This commands forces the sensor to perform the Remote Teach operation on the next trigger. This command can be performed using the sensor's touchscreen.

Examples

```
>>  do teach\x0D\x0A
<<  OK\x0D\x0A
```

4.4.7 ProductChange Command Group

Command	Group	Item	Description
Do	ProductChange	[*Name*]	Forces the sensor to switch to the specified inspection. The sensor does not transmit a response until the sensor has completed the action. Inspections results will be invalid until the next trigger.
Get	ProductChange	InspectionNames	List of all inspections stored in the sensor.

Examples

```
>>  get productchange inspectionnames\x0D\x0A
<<  OK\x0D\x0A
```

```
>>  do productchange "inspection2"\x0D\x0A
<<  OK\x0D\x0A
```

4.4.8 History Command Group

Command	Group	Item	Description
Get	History	Passed	The number of passed inspections.
Get	History	Failed	The number of failed inspections.
Get	History	MissedTriggers	The number of missed triggers.
Get	History	TotalFrames	The total number of inspections since the history was last cleared.
Get	History	MinInspectionTime	The minimum elapsed time (msec) of the inspection.
Get	History	MaxInspectionTime	The maximum elapsed time (msec) of the inspection.
Do	History	Clear	Clears all history fields (for example pass, fail, sensor history, etc.).

Examples

```
>>  get history passed\x0D\x0A
<<  OK\x0D\x0A
<<  13\x0D\x0A
```

```
>>  get history startframenumber\x0D\x0A
<<  OK\x0D\x0A
<<  3\x0D\x0A
```

```
>>  do history clear\x0D\x0A
<<  OK\x0D\x0A
```

4.4.9 Inspection Command Group

Command	Group	Item	Description
Get	Inspection	Status	This status of the most recent inspection either Pass, Fail, or Idle (no triggers).

Command	Group	Item	Description
Get	Inspection	Name	The name of the active inspection.
Get	Inspection	FrameNumber	The most recent inspection frame number.
Get	Inspection	ExecutionTime	The most recent inspection execution time in msec.

Examples

```
>>  get inspection status\x0D\x0A
<<  OK\x0D\x0A
<<  Fail\x0D\x0A
```

```
>>  get inspection executiontime\x0D\x0A
<<  OK\x0D\x0A
<<  37.739\x0D\x0A
```

4.4.10 AREA_RESULT Command Group

Command	Group	Item	Description
Get	AREA_RESULT	Count	The number of detected areas.
Get	AREA_RESULT	MinArea	The size of the smallest detected area.
Get	AREA_RESULT	MaxArea	The size of the largest detected area.

Examples

```
>>  get area_result count\x0D\x0A
<<  OK\x0D\x0A
<<  2\x0D\x0A
```

```
>>  get area_result minarea\x0D\x0A
<<  OK\x0D\x0A
<<  7665\x0D\x0A
```

4.4.11 AREA_HISTORY Command Group

Command	Group	Item	Description
Get	AREA_HISTORY	MinCount	The minimum number of detected areas, since history was last cleared.
Get	AREA_HISTORY	MaxCount	The maximum number of detected areas, since history was last cleared.
Get	AREA_HISTORY	MinArea	The minimum detected area value, since history was last cleared.
Get	AREA_HISTORY	MaxArea	The maximum detected area value, since history was last cleared.

Examples

```
>>  get area_history mincount\x0D\x0A
<<  OK\x0D\x0A
<<  1\x0D\x0A
```

```
>>  get area_history minarea\x0D\x0A
<<  OK\x0D\x0A
<<  7665\x0D\x0A
```

4.4.12 BLEMISH_RESULT Command Group

Command	Group	Item	Description
Get	BLEMISH_RESULT	Count	The number of detected blemishes.
Get	BLEMISH_RESULT	MinEdgeLength	The minimum detected blemish edge length.
Get	BLEMISH_RESULT	MaxEdgeLength	The maximum detected blemish edge length.

Examples

```
>>  get blemish_result count\x0D\x0A
<<  OK\x0D\x0A
<<  4\x0D\x0A
```

```
>>  get blemish_result minedgelength\x0D\x0A
<<  OK\x0D\x0A
<<  22\x0D\x0A
```

4.4.13 BLEMISH_HISTORY Command Group

Command	Group	Item	Description
Get	BLEMISH_HISTORY	MinCount	The minimum number of detected blemishes, since history was last cleared.
Get	BLEMISH_HISTORY	MaxCount	The maximum number of detected blemishes, since history was last cleared.
Get	BLEMISH_HISTORY	MinEdgeLength	The minimum detected blemish edge length, since history was last cleared.
Get	BLEMISH_HISTORY	MaxEdgeLength	The maximum detected blemish edge length, since history was last cleared.

Examples

```
>>  get blemish_history count\x0D\x0A
<<  OK\x0D\x0A
<<  1\x0D\x0A
```

```
>>  get blemish_history maxcount\x0D\x0A
<<  OK\x0D\x0A
<<  6\x0D\x0A
```

4.4.14 MATCH_RESULT Command Group

Command	Group	Item	Description
Get	MATCH_RESULT	Count	The number of detected matches.
Get	MATCH_RESULT	MinPercentMatch	The minimum detected match percentage.
Get	MATCH_RESULT	MaxPercentMatch	The maximum detected match percentage.

Examples

```
>>  get match_result count\x0D\x0A
<<  OK\x0D\x0A
<<  1\x0D\x0A
```

```
>>  get match_result maxpercentmatch\x0D\x0A
<<  OK\x0D\x0A
<<  6\x0D\x0A
```

4.4.15 MATCH_HISTORY Command Group

Command	Group	Item	Description
Get	MATCH_HISTORY	MinCount	The minimum number of detected matches, since history was last cleared.
Get	MATCH_HISTORY	MaxCount	The maximum number of detected matches, since history was last cleared.
Get	MATCH_HISTORY	MinPercent	The minimum detected match percentage, since history was last cleared.
Get	MATCH_HISTORY	MaxPercent	The maximum detected match percentage, since history was last cleared.

Examples

```
>>  get match_history count\x0D\x0A
<<  OK\x0D\x0A
<<  1\x0D\x0A
```

```
>>  get match_history maxcount\x0D\x0A
<<  OK\x0D\x0A
<<  6\x0D\x0A
```

4.4.16 SORT_RESULT Command Group

Command	Group	Item	Description
Get	SORT_RESULT	Count	The number of detected sort patterns.
Get	SORT_RESULT	MinPercentMatch	The minimum detected sort pattern match percentage.
Get	SORT_RESULT	MaxPercentMatch	The maximum detected sort pattern match percentage.
Get	SORT_RESULT	PatternNumbers	Listing of detected patterns by pattern number.
Get	SORT_RESULT	PatternNames	Listing of detected patterns by pattern name.

4.4.17 SORT_HISTORY Command Group

Command	Group	Item	Description
Get	SORT_HISTORY	MinCount	The minimum number of detected sort patterns, since history was last cleared.
Get	SORT_HISTORY	MaxCount	The maximum number of detected sort patterns, since history was last cleared.
Get	SORT_HISTORY	MinPercent	The minimum detected sort pattern match percentage, since history was last cleared.
Get	SORT_HISTORY	MaxPercent	The maximum detected sort pattern match percentage, since history was last cleared.

Examples

```
>>  get sort_history mincount\x0D\x0A
<<  OK\x0D\x0A
<<  1\x0D\x0A
```

```
>>  get sort_history maxcount\x0D\x0A
<<  OK\x0D\x0A
<<  6\x0D\x0A
```

4.5 Multiple Sensors Inspection

When having more than one sensor in the inspection, all SensorType_Result and SensorType_History related commands must include the *Sensor Name* in "< >" brackets.

Examples

```
>> get area_result <Area1> count\x0D\x0A
<< OK\x0D\x0A
<< 7665\x0D\x0A
```

Notice that <Area1> is the sensor name of an Area Sensor Type in the current inspection.

```
>> get blemish_history <Blemish1> minedgelength\x0D\x0A
<< OK\x0D\x0A
<< 22\x0D\x0A
```

Notice that <Blemish1> is the sensor name of a Blemish Sensor Type in the current inspection.

4.6 Command Channel Command Status Register

The command status is a verification of the command type.

Command Status	Value (16-bit integer)
Unknown	0
Read	1
Write	2
Execute	3

4.7 Command Channel Error Codes

Table 3: Plus TG Command Channel Error Codes

Numeric ID	Text ID	Description
00000	SUCCESS	Command processed successfully
10000	EMPTY_FRAME_RECEIVED	Indicates that the request was empty. The command channel requires a command, any arguments, and an end-of-frame delimiter.
10001	COMMAND_NOT_RECOGNIZED	The command specified is not recognized
10100	GROUP_MISSING	A Group ID must be specified immediately after the command
10101	GROUP_NOT_FOUND	The specified Group ID is invalid / unknown
10102	GROUP_ITEM_MISSING	A Group Item ID must be specified immediately after the Group ID

Numeric ID	Text ID	Description
10103	GROUP_ITEM_NOT_FOUND	The specified Group Item ID is invalid / unknown
10152	NOT_READABLE	Attempt to get a value that is not readable
10153	NOT_WRITEABLE	Attempt to set a value that is not writeable
10250	NOT_A_METHOD	Method ID specified is not a method
10251	WRONG_ARGUMENT_COUNT	Total method arguments specified do not match method
10252	COMMAND_NOT_FINISHED	Attempt to issue command when a previous command has not finished
10300	INVALID_ARGUMENT_TYPE	Item ID specified must be a item (not a group or method)
10301	DATA_VALUE_MISSING	Command missing item's data value
10350	ARGUMENTS_DETECTED	Get command received with unneeded arguments
10351	INVALID_ARGUMENT_TYPE	Item ID specified must be a item (not a group or method)
10340	MINIMUM_VALUE_EXCEEDED	New item value is below the minimum
10341	MAXIMUM_VALUE_EXCEEDED	New items value is above the maximum
10500	DATA_SET_EMPTY	Data export operation returned no results.
10900	SENSOR_NOT_READY	Command specified requires sensor to be in the READY state.
10920	SENSOR_TYPE_NOT_ACTIVE	Command specified belongs to a different sensor type.
15000	VALUE_INVALID	Text value is invalid / unknown
15050	VALUE_INVALID	Text value is invalid - expecting True or False
15100	STRING_TOO_LONG	String value specified exceeds maximum allowable length
20200	NO_AREAS_FOUND	Attempt to obtain value when no areas were found.
20600	NO_MATCHES_FOUND	Attempt to obtain value when no matches were found.
20800	NO_MATCHES_FOUND	Attempt to obtain value when no sort patterns were found.
80000	REMOTE_DISPLAY_NOT_CONNECTED	Remote Display must be connected to obtain this value
80001	REMOTE_DISPLAY_NOT_SUPPORTED	This sensor does not have Remote Display capability
80100	COMMAND_MODE_EXPECTED	The Trigger Mode must be set to "Command" perform this operation
80101	COMMAND_TIMED_OUT	The command timed out before finishing
80102	TRIGGER_REQUIRED	Access to the specified data requires a triggered inspection
80150	COMMAND_TIMED_OUT	The command timed out before finishing
80200	SYSTEM_ERROR_NOT_ACTIVE	The System Error must be active to execute this command
80300	TEACH_SENSOR_TYPE_INVALID	Teach requires Match Sensor type.
80350	MULTIPLE_INSPECTIONS_DISABLED	Requires multiple inspections to be enabled
80351	MULTIPLE_INSPECTIONS_EMPTY	No inspections are available in multiple inspection mode.
80400	PRODUCT_CHANGE_WHEN_NOT_READY	Sensor must be in the READY state to perform a product change.
80401	PRODUCT_CHANGE_INVALID_INSPECTION	Attempt to product change to a unknown or invalid inspection.
80402	PRODUCT_CHANGE_TIMEOUT	The Product Change operation timed out.
80403	PRODUCT_CHANGE_TO_SAME_INSPECTION	Attempt to product change to the same inspection.
80404	SENSOR_NAME_NOT_FOUND	Attempt to use a command without a sensor name in a multi-sensor inspection

Index