

# **Bristol IP Driver**

**©2015 Kepware, Inc.**

# Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
Bristol IP Driver Help .....	5
Overview .....	5
BSAP Networks .....	5
<b>Channel Setup</b> .....	<b>7</b>
<b>Device Setup</b> .....	<b>8</b>
Communication .....	8
Timing .....	9
Tag Import Settings .....	10
<b>Optimizing Communications</b> .....	<b>11</b>
<b>Data Types Description</b> .....	<b>12</b>
<b>Address Descriptions</b> .....	<b>13</b>
<b>Automatic Tag Generation</b> .....	<b>14</b>
<b>Error Descriptions</b> .....	<b>17</b>
<Operation> on device <device name> signal <signal> caused an overflow. Some data has been lost. ....	17
<Operation> on device <device name> signal <signal> failed. The device does not contain this signal. ....	18
<Operation> on device <device name> signal <signal> failed. The device requires higher level permissions to perform this operation. ....	18
<Operation> on device <device name> signal <signal> failed. The request was malformed or incomplete. ....	18
A read by address failed because the MSD version for device <device name> changed. The driver is sending read by name requests to get the updated MSD version from the device. ....	18
A write by address failed because the MSD version for device <device name> changed. The driver is sending a write by name request to the device. ....	19
Failed to send a request to device <device>. The device is not communicating. ....	19
Failed to send a request to device <device>. The packet is malformed or too large. ....	19
Failed to stop Winsock communications. ....	20
Failure to initiate 'winsock.dll' [Error: <error>]. ....	20
Invalid BSAP function. ....	20
Read failed on device <device> address <address>. The server expects this address to be of type <type>, but it is of type <type> in the device. ....	21
The response received from device <device> contains an invalid sequence number. Review timing properties. ....	21
The response received from device <device> is malformed. ....	21
The response received from device <device> is too large (%<count> bytes). The data is lost. ....	22
The signal was not found. ....	22
The string written to device <device>, address <address> is too long. The maximum supported string is <count> characters. The write will continue as is, but the device will truncate the value. ....	22
Unable to bind to port <port> on Channel <channel>. ....	22
Unable to create a socket connection for Channel <channel>. ....	23

---

Unable to generate a tag database for device <device name>. Reason: Import file does not contain any valid signals. ....	23
Write to device <device name> signal <signal> failed. The signal's manual inhibit attribute is set to false, not allowing writes. ....	23
Write to device <device name> signal <signal> failed. The signal is read only. ....	23
Write to device <device name> signal <signal> failed. The write value is invalid. Make sure the value is within the approved BSAP range. ....	24
<b>Index</b> .....	<b>25</b>



## Bristol IP Driver Help

---

Help version 1.014

### CONTENTS

#### [Overview](#)

What is the Bristol IP Driver?

#### [Channel Setup](#)

How do I configure channels for use with this driver?

#### [Device Setup](#)

How do I configure a device for use with this driver?

#### [Optimizing Communications](#)

How do I optimize communications on the Bristol IP Driver?

#### [Data Types Description](#)

What data types are supported by this driver?

#### [Address Descriptions](#)

How do I address a data location on the Bristol IP Driver?

#### [Automatic Tag Generation](#)

How can I easily configure tags for this driver?

#### [Error Descriptions](#)

What error messages are produced by the Bristol IP Driver?

### Overview

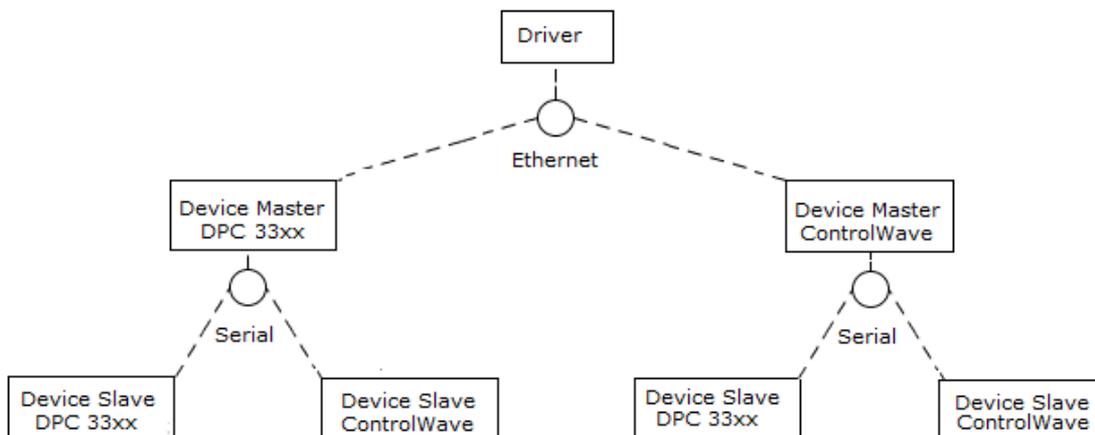
---

The Bristol IP Driver provides a reliable way to communicate with DPC and ControlWave devices. It supports communications to both Ethernet-enabled Bristol devices and to serial Bristol devices (through an Ethernet-enabled Bristol device acting as a gateway). It supports Network 3000 Series (DPC 33xx) and ControlWave devices. For more information, refer to [Device Setup](#).

### BSAP Networks

---

The image below displays a typical Bristol IP Driver configuration. The devices are physical devices. The server channels and devices are not displayed for simplicity.



The Bristol IP Driver communicates to Master or Slave devices using Ethernet (UDP). Communications to Master devices are direct: the driver sends a request to the device and it responds. Communications to Slave devices are indirect: the driver sends a request through the Master device. In most configurations, Slave devices communicate over serial to the Master.

**Note:** BSAP allows six levels of hierarchy including the Master device, or five levels of hierarchy below the Master device. Each level below the Master device can have a maximum of 128 devices.

**Operations By Symbolic Name vs. By Address**

Bristol IP devices support reads and writes by address or by symbolic name. Reads and writes by symbolic name are usually less efficient because the request contains the signal name in ASCII format. Reads and writes by address are usually more efficient because the address is only two bytes. The driver will request by address whenever possible.

**Supported BSAP Functions**

The Bristol IP Driver supports a set of Bristol BSAP functions that are a subset of those available in the Remote Database Access (RDB) specification. All request types have the ability to request more than one signal at a time. For efficiency, the driver will include multiple signals in a request whenever possible.

**Note:** Requests can be made using either the signal's symbolic name or the signal's address (which is generally more efficient).

Operation	Description
Read Signal By Address	Read one or more signals using the two byte address.
Read Signal By Symbolic Name	Read one or more signals by signal name. Request the signal's two byte address in addition to the value.
Write Signal By Address	Write one or more signals using the two byte address.
Write Signal By Symbolic Name	Write one or more signals by signal name.

## Channel Setup

---

### Network Interface

The Network Interface dialog specifies the Network Interface Controller (NIC) over which the channel will communicate. Each channel must have a unique NIC and port. For example, two Bristol channels cannot be bound to the default NIC on Port 1234: only one channel can bind to the port and communicate.

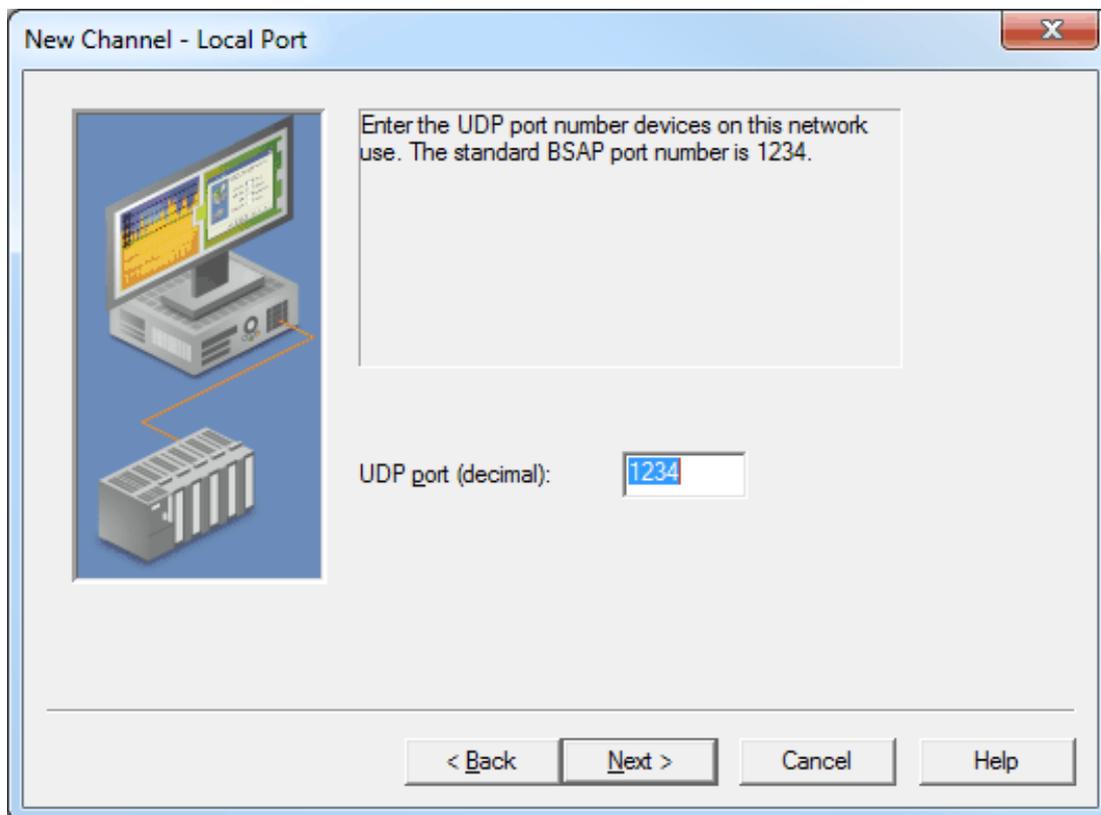
**Note:** The default adapter is the adapter bound to the machine hostname.

### Communication Serialization

The Bristol IP Driver supports Communication Serialization, which specifies whether data transmissions should be limited to one channel at a time. For more information, refer to "Channel Properties - Advanced" in the server help file.

### Local Port

The Local Port dialog specifies the port that all devices under the channel will use for communications.



Description of the parameter is as follows:

- **UDP Port:** This parameter specifies the local port to which the channel will be bound. The default setting for most Bristol devices is 1234. Users that change the UDP Port may also need to change the device configuration to communicate with the port.

**Note:** This driver supports additional channel settings. For more information, refer to "What is a Channel?" in the server help file.

## Device Setup

### Supported Devices

- Network 3000 Series (DPC 33xx) devices with BSAP IP (Ethernet) support. 386EX Protected Mode DPC3330 and DPC 3335 with PES03/PEX03 or newer Firmware is required.
- ControlWave devices with BSAP IP (Ethernet) support. ControlWave Micro CWP/LPS/CWR 02.00 or newer Firmware is required.

### Communication Protocol

Bristol BSAP IP Protocol (UDP)

### Maximum Number of Channels and Devices

The maximum number of channels supported by this driver is 256. The maximum number of devices supported is 8192.

## Communication

The Communication dialog specifies the device's communication settings.

**New Device - Communication**

Specify the Global Address of the RTU. For local devices, enter 0.

Specify the IP Address of the RTU.

Specify the UDP port of the RTU.

Specify the maximum number of bytes per each read or write request.

RTU Global Address (hex):

RTU IP Address:

RTU UDP Port Number (dec):

Maximum Bytes Per Request:

< Back    Next >    Cancel    Help

Descriptions of the parameters are as follows:

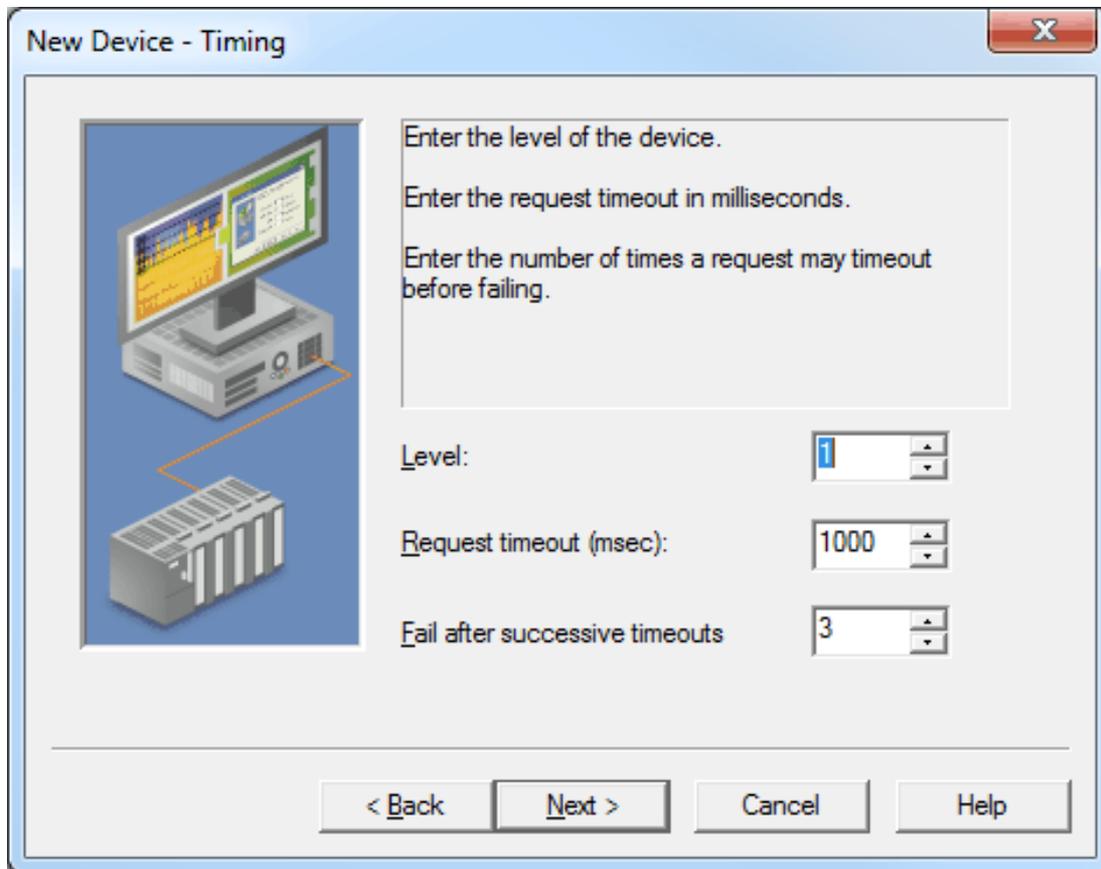
- **RTU Global Address:** This parameter specifies the hex address of the device. For Master devices, the default setting is 0. For Slave devices, the default setting is non-zero. To find the address in OpenBSI NetView, right-click on the device and select **Properties | BSAP**.
- Note:** Although Master devices can also have a non-zero global address, users should follow convention and use the zero global address for communications.
- **RTU IP Address:** This parameter specifies the IP address of the RTU. For Master devices, this is the IP of the Master. For Slave devices, this is the IP of the Master.
- Note:** In the BSAP IP protocol, the Master device passes requests to the Slaves using the Global Address.

- **RTU UDP Port Number:** This parameter specifies the port number of the Master device. For Slave devices, this is the port number of the Master.
- **Maximum Bytes Per Request:** This parameter specifies the maximum number of bytes that will be sent to the device per request. The amount specified should depend on the device's bandwidth and performance requirements.

**Note:** If communications with a device are fast and bandwidth is available, users should specify a larger number of bytes per request. If communications are slow or bandwidth is limited, users should specify a smaller amount of bytes per request.

## Timing

The Timing dialog controls the device's timeout settings.



Descriptions of the parameters are as follows:

- **Level:** This parameter specifies the device's network level. The valid range is 1 to 6. The default setting is 1.

**Note:** Level One devices are Masters and have a Global Address of 0. Level Two devices are Slaves. Level Three devices are Slaves of Slaves. The higher the level, the more latency when communicating with the device. This setting updates the "Request timeout" parameter.

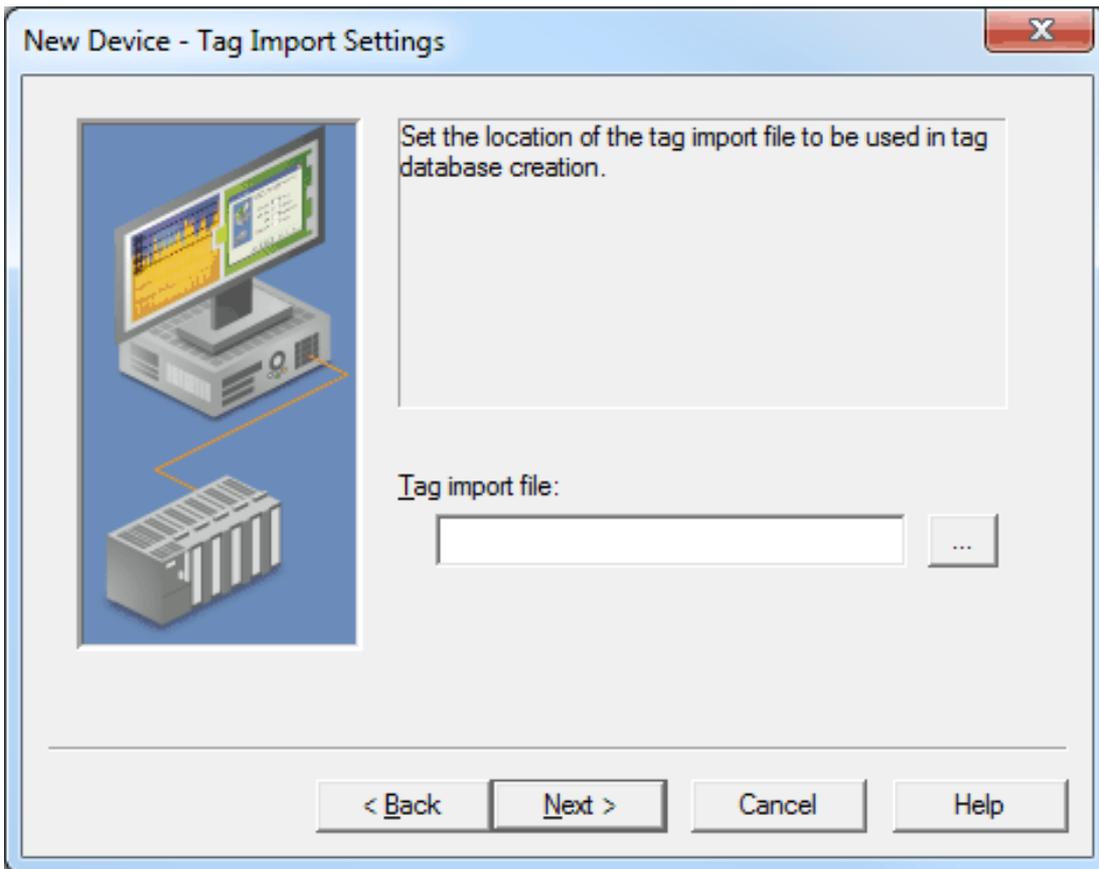
- **Request timeout:** This parameter specifies the number of milliseconds that the driver will wait for a response from the device. The default setting is 1000 milliseconds.

**Note:** If a response is not received within this period, the driver will resend the request depending on the "Fail after successive timeouts" parameter. If the driver receives a malformed response, it will resend the request right away. Failure recovery improves when the device sends a valid response that is corrupted by network noise.

- **Fail after successive timeouts:** This parameter specifies how many request timeouts can occur before the request is failed. The valid range is 1 to 10. The default setting is 3.

## Tag Import Settings

The Tag Import Settings dialog specifies the tag import file that will be used in automatic tag generation.



Description of the parameter is as follows:

- **Tag import file:** This parameter specifies the device's tag import file. It must be a valid signal (\*.sig) file. For more information, refer to [Automatic Tag Generation](#).

## Optimizing Communications

---

This driver supports communications with underlying BSAP child devices through a Master Ethernet device. Due to the nature of the BSAP protocol, communications to lower-level BSAP devices may be slower. As such, users may need to make special design considerations in the server configuration to attain optimal communication performance. For more information on improving performance when communicating with slower devices, refer to the subtopics below.

### Channel and Device Setup

Communication to multiple devices on a channel is serialized. When communications with one device is slow, the performance of the other devices on the channel will decrease. To avoid degradation in performance, users can place slower devices on separate channels.

**Note:** Devices located on a common BSAP level should be grouped under the same channel in the server.

### Multi-Home NIC

Separating devices onto different channels effectively improves performance, but is limited in that each channel requires a unique port and NIC combination. To eliminate the limitation, users can multi-home an NIC card and bind multiple channels to the same port.

### Request Size

The device's Request Size directly impacts performance. For maximum performance, 256 bytes is generally recommended; however, a lower request size may increase performance in situations where bandwidth is limited. Lower request sizes are also generally recommended for non-Master devices. The valid range is 64 to 256 bytes.

### Request Frequency

Frequently asking slower devices for data will slow down a channel's communications. To better control this, users can limit the rate at which data is requested from the device with the Scan Mode setting.

### Cached Data

The Bristol IP Driver does not maintain cached tag data after the client removes the tag from the server. Every time the server adds or re-adds a tag on behalf of the requesting client, the server will request the data from the device and send it to the client. This behavior ensures data freshness. Applications that require the reading of cached values should do so at the client level. Techniques for caching are as follows:

1. Have the client add the tag as inactive. Then, perform reads when fresh data is needed.
2. Have the client add the tag as active and configure the device's Scan Mode setting to poll the device. Then perform cached reads when data is needed.

## Data Types Description

---

Data Type	Description
Boolean	1 byte
Float	4-byte floating point
String	Zero-terminated ASCII character array

## Address Descriptions

---

Bristol addresses depend on the model. For more information, refer to the subtopics below.

### Network 3000 Series Devices

The syntax for Network 3000 Series devices is as follows: *<Base>.<Ext>.<Att>*.

- **Base:** This element must be 1 to 8 characters in length and begin with a letter or pound sign (#). The remaining characters will consist of letters and numbers.
- **Ext:** This element must be 0 to 6 characters in length, and consist of letters or numbers.
- **Att:** This element must be 0 to 4 characters in length, and consist of letters or numbers.

**Note:** An example of the syntax is "#DIAL.001.Val".

### ControlWave Devices

The syntax for ControlWave devices is as follows: *<Access>.<Signal Name>*.

- **Access:** This element is the Bristol program name for local variables (or "%GV" for global and system variables).
- **Signal Name:** This element must contain letters, numbers, or a period and start with a letter or underscore. Signal names that start with an underscore are system variables.

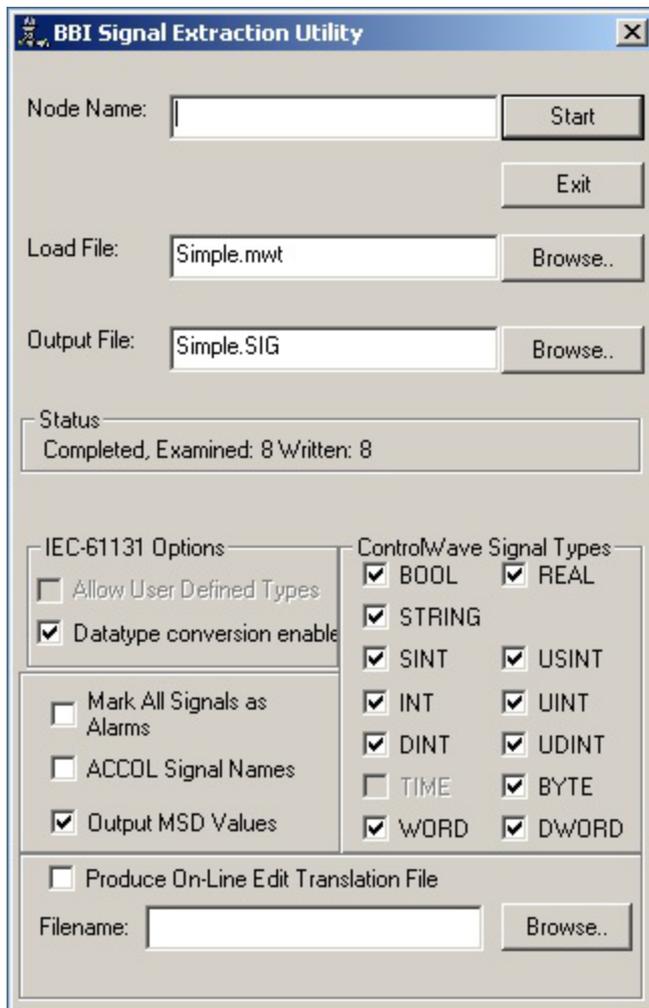
**Note:** An example of the syntax is "%GV.MYSIGNAL".

## Automatic Tag Generation

The Bristol IP Driver can automatically generate tags using a signal (\*.sig) file, which is created by Bristol OpenBSI software. The procedure for creating the signal file is different for DPC 33xx devices and ControlWave devices.

### DPC33xx Devices

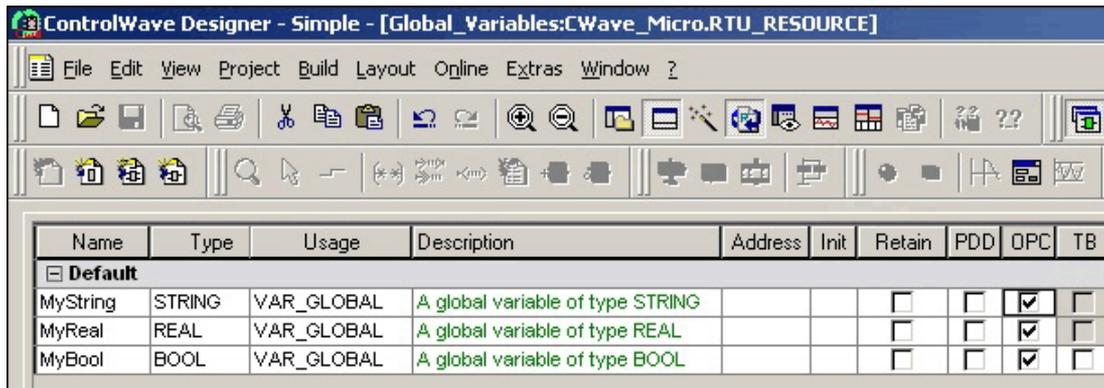
1. To start, open the **ACCOL Workbench**. Then, define a signal.
2. In **Signal Properties**, ensure that **Mark as Global** is checked.
3. Next, launch the **BBI Signal Extraction Utility**. Then, locate **Mark All Signals as Alarms** and ensure that it is unchecked.
4. In **Load File**, browse for and specify the load file.
5. In **Output File**, browse for and specify the output file.



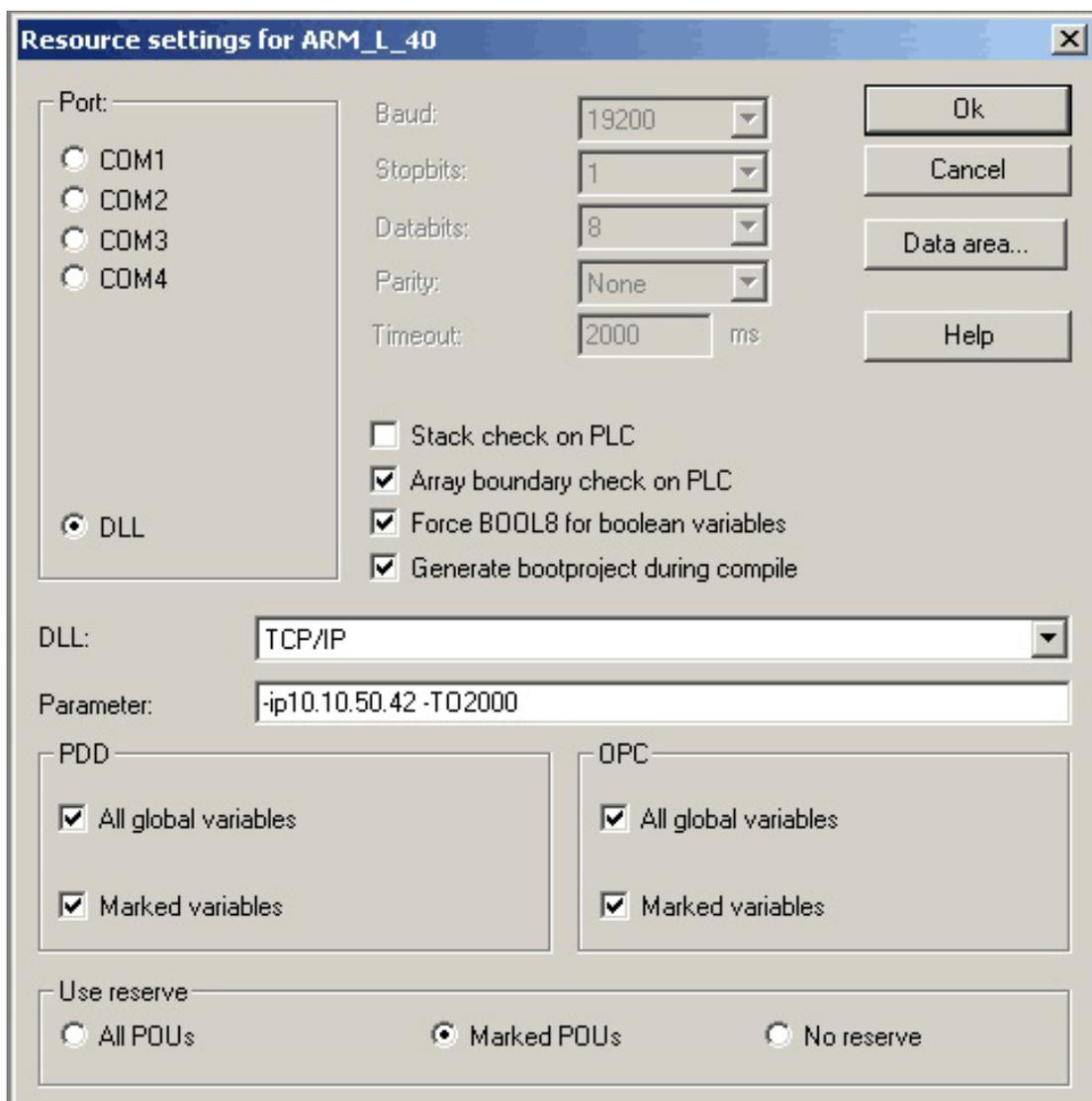
6. Once finished, click **Start** to extract the signals.

### ControlWave Devices

1. To start, open **ControlWave Designer**. Then, ensure that the **OPC** column check box is checked for all signals that will be exported.



- Next, open the controller's **Resource Settings**.



- Locate **PDD**, then ensure that the **Marked Variables** option is checked.

**Note:** All Global Variables is optional.

- Once finished, click **OK**.
- Choose **Start|All Programs|OpenBSI Tools|Common Tools|Signal Extractor**.

6. Select the **Browse** (...) button next to the "Load File:" and locate and select the .mwt project with the tags
7. If desired, select a different output directory by using the **Browse** (...) button next to the "Output File:" and locate the desired location
8. Select **Start** and the tag file is exported to the selected folder.
9. Place the exported signal file where the server can access it.
10. Enter the signal file into the "Tag import file" field in the **Tag Import Settings** tab of the device properties.
11. The signals can now be created with the database creation functionality.

## Error Descriptions

---

The following messages may be generated. Click on the link for a description of the message.

[<Operation> on device <device name> signal <signal> caused an overflow. Some data has been lost.](#)

[<Operation> on device <device name> signal <signal> failed. The device does not contain this signal.](#)

[<Operation> on device <device name> signal <signal> failed. The device requires higher level permissions to perform this operation.](#)

[<Operation> on device <device name> signal <signal> failed. The request was malformed or incomplete.](#)

[A read by address failed because the MSD version for device <device name> changed. The driver is sending read by name requests to get the updated MSD version from the device.](#)

[A write by address failed because the MSD version for device <device name> changed. The driver is sending a write by name request to the device.](#)

[Failed to send a request to device <device>. The device is not communicating.](#)

[Failed to send a request to device <device>. The packet is malformed or too large.](#)

[Failed to stop Winsock communications.](#)

[Failure to initiate 'winsock.dll' \[Error:<error>\].](#)

[Invalid BSAP function.](#)

[Read failed on device <device> address <address>. The server expects this address to be of type <type>, but it is of type <type> in the device.](#)

[The response received from device <device> contains an invalid sequence number. Review timing properties.](#)

[The response received from device <device> is malformed.](#)

[The response received from device <device> is too large \(%<count> bytes\). The data is lost.](#)

[The signal was not found.](#)

[The string written to device <device>, address <address> is too long. The max supported string is <count> characters. The write will continue as is, but the device will truncate the value.](#)

[Unable to bind to port <port> on Channel <channel>.](#)

[Unable to create a socket connection for Channel <channel>.](#)

[Unable to generate a tag database for device <device name>. Reason: Import file does not contain any valid signals.](#)

[Write to device <device name> signal <signal> failed. The signal's manual inhibit attribute is set to false, not allowing writes.](#)

[Write to device <device name> signal <signal> failed. The signal is read only.](#)

[Write to device <device name> signal <signal> failed. The write value is invalid. Make sure the value is within the approved BSAP range.](#)

**<Operation> on device <device name> signal <signal> caused an overflow. Some data has been lost.**

---

### Error Type:

Error

### Possible Cause:

A read or write operation on a signal caused a data overflow. The operation completed, but only part of the data was read or written to the device.

### Solution:

1. If writing data to the device, verify that the data is in the correct format for the data type.
2. Verify that the String being written to the device is within the 64 character limit and is null terminated.

### Note:

Boolean Tags take up one byte, Float Tags take up four bytes, and String Tags can be up to 64 characters (and must be null terminated).

---

**<Operation> on device <device name> signal <signal> failed. The device does not contain this signal.**

---

**Error Type:**

Error

**Possible Cause:**

1. An operation was performed on a signal that is not in the device.
2. An attempt was made to write an out-of-range value to a tag in a ControlWave device.

**Solution:**

1. Check the tag address, and ensure that it matches the signal name in the device.
2. Check the data type of the tag in the ControlWave device. Then, ensure the value that will be written is within the range of values for that data type.

---

**<Operation> on device <device name> signal <signal> failed. The device requires higher level permissions to perform this operation.**

---

**Error Type:**

Error

**Possible Cause:**

The device does not have sufficient permissions to perform the operation.

**Solution:**

Check the permission settings for the signal in the device configuration. Then, ensure that the signal allows the operation to be performed with the highest level of permissions.

**Note:**

The permission level is sent by the driver as part of the request, and is set to the highest level by default. This error is uncommon.

---

**<Operation> on device <device name> signal <signal> failed. The request was malformed or incomplete.**

---

**Error Type:**

Error

**Possible Cause:**

1. The driver is attempting to communicate with a device that is not supported.
2. Communications between the driver and device are noisy and causing malformed packets.

**Solution:**

1. Ensure that the device with which the driver is communicating is supported.
2. Ensure that the communication line between the driver and device is reliable.

**Note:**

This error may indicate a problem in the driver. If the solutions listed above fail, contact Technical Support.

---

**A read by address failed because the MSD version for device <device name> changed. The driver is sending read by name requests to get the updated MSD version from the device.**

---

**Error Type:**

Warning

**Possible Cause:**

A configuration change occurred in the device that caused the MSD version to update. This changed the MSD address for all signals and caused the read to fail.

**Solution:**

The driver will reissue the read as a read-by-name and will automatically update the MSD information. No action is necessary.

**Note:**

It is not recommended that configuration changes be made to the physical device while the driver is connected and receiving data. Users should disconnect the driver, make the configuration changes, and reconnect.

**A write by address failed because the MSD version for device <device name> changed. The driver is sending a write by name request to the device.****Error Type:**

Warning

**Possible Cause:**

A configuration change occurred in the device that caused the MSD version to update. This changed the MSD address for all signals and caused the write to fail.

**Solution:**

The driver will reissue the write as a write-by-name. This will complete the write, but will not update the device's MSD information. Future writes will be write-by-name until a read occurs.

**Note:**

It is not recommended that configuration changes be made to the physical device while the driver is connected and receiving data. Users should disconnect the driver, make the configuration changes, then reconnect.

**Failed to send a request to device <device>. The device is not communicating.****Error Type:**

Error

The communication timed out.

**Possible Cause:**

1. The device is offline or the driver is not configured to communicate with the device properly.
2. The device is not supported.

**Solution:**

1. Ensure that the device can be reached by the driver by attempting to ping the device's IP address.
2. Ensure that the device is supported.

**Failed to send a request to device <device>. The packet is malformed or too large.****Error Type:**

Error

**Possible Cause:**

1. The driver is attempting to communicate with a device that is not supported.
2. Communications between the driver and device are noisy and are causing malformed packets.

3. The driver is attempting to write to a tag whose address length plus data length exceeds the maximum message size that is supported by the Bristol BSAP IP protocol.

**Solution:**

1. Ensure that the device with which the driver is communicating is supported.
2. Ensure that the communication line between the driver and device is reliable.
3. Perform a device read on the tag, then attempt a write. If the error persists, shorten the address in the device and then try again.

**Note:**

This error may indicate a problem in the driver. If the solutions listed above fail, contact Technical Support.

**Failed to stop Winsock communications.**

---

**Error Type:**

Error

Winsock was placed in an invalid state. This may cause the next startup to fail.

**Possible Cause:**

1. Failed to cleanup Winsock while the driver was shutting down.
2. Winsock was never initialized.
3. A blocking socket call was in progress while the driver was shutting down.

**Solution:**

1. Restart the server and connect to a Bristol device.
2. Ensure that the operating system has the latest patches.

**Failure to initiate 'winsock.dll' [Error:<error>].**

---

**Error Type:**

Error

The driver cannot communicate over Ethernet without Winsock.

**Possible Cause:**

1. The underlying system is not ready to initialize network communications.
2. The version of Winsock requested is not supported.
3. A Winsock operation is already in progress.

**Solution:**

1. Restart the server.
2. Ensure that the operating system has the latest patches.

**Invalid BSAP function.**

---

**Error Type:**

Error

**Possible Cause:**

1. The BSAP function code sent to the device is invalid. The operation failed.
2. The device is not officially supported. The operation failed.
3. Communications between the driver and device are unreliable. The operation failed.

**Solution:**

1. Ensure that the device is supported.
2. Ensure that the communication line between the driver and device is reliable.

**Read failed on device <device> address <address>. The server expects this address to be of type <type>, but it is of type <type> in the device.****Error Type:**

Error

**Possible Cause:**

The client requested a Float Tag, but the actual type is a String. The tag read failed and the tag was set to Bad quality.

**Solution:**

For Static Tags, match the client-requested type to the device type by changing the tag type in the configuration. For Dynamic Tags, the client must change the request type.

**Note:**

This error can occur any time that the client type does not match the device type. The device read succeeds, but the server cannot convert the device type to the client type.

**The response received from device <device> contains an invalid sequence number. Review timing properties.****Error Type:**

Warning

**Possible Cause:**

A response was received from the device, but the sequence number in the response packet does not match the sequence number issued in the request. The response will be ignored. The driver will continue waiting for a valid response (depending on the Timing settings specified in Device Properties).

**Solution:**

1. Increase the value specified in the device's Request Timeout parameter.
2. Increase the value specified in the device's Fail After Successive Timeouts parameter.

**The response received from device <device> is malformed.****Error Type:**

Error

**Possible Cause:**

1. The device is not supported. The response will be ignored. A new request will be sent, depending on the device's timeout configuration.
2. The communications line between the driver and the device is unreliable.

**Solution:**

1. Ensure that the device is supported.
2. Verify that the communication line between the driver and device is reliable.

---

**The response received from device <device> is too large (%<count> bytes).  
The data is lost.**

---

**Error Type:**

Error

**Possible Cause:**

The size of the device's response is too large for the driver. The driver ignored the response.

**Solution:**

1. Ensure that the device with which the driver is communicating is supported.
2. Limit the size of the requests by changing the Maximum Bytes Per Request setting. This will also limit the size of the response.

**Note:**

This error is uncommon.

---

**The signal was not found.**

---

**Error Type:**

Error

**Possible Cause:**

The signal was not in the device. The operation on the signal failed.

**Solution:**

Check the device configuration and ensure that the signal is in the device and matches the tag address.

---

**The string written to device <device>, address <address> is too long. The maximum supported string is <count> characters. The write will continue as is, but the device will truncate the value.**

---

**Error Type:**

Warning

**Possible Cause:**

The String being written to the device is too large. The device ignored the excess characters in the String or ignored the write entirely.

**Solution:**

Check the maximum String size allowed for the address in the device. Then, ensure that the write does not exceed this size.

**Note:**

The default String size is 64 characters (including the null terminator).

---

**Unable to bind to port <port> on Channel <channel>.**

---

**Error Type:**

Error

**Possible Cause:**

The port is already in use or is unavailable. There are no communications on the channel.

**Solution:**

Verify that no other processes are using the port and Network Interface Controller. To do so, run the "netstat" command in Windows. Then, change the port or network interface if possible.

**Note:**

This error is most common in cases where channels have duplicate Network Interface Controllers and ports. This can occur between any channels; it is not limited to channels for this driver. Other programs on the machine can also be using the port.

**Unable to create a socket connection for Channel <channel>.**

---

**Error Type:**

Error

**Possible Cause:**

While initializing a channel, the driver failed to get a valid socket from the system. There are no communications on the channel.

**Solution:**

Restart the server or reconnect the clients.

**Note:**

This failure usually indicates that there is something wrong with the system (which could be out of sockets or in a Bad state).

**Unable to generate a tag database for device <device name>. Reason: Import file does not contain any valid signals.**

---

**Error Type:**

Error

**Possible Cause:**

A corrupted .sig file was imported. The tag database was not generated.

**Solution:**

Verify that the .sig file is valid. Then, re-import the file.

**Write to device <device name> signal <signal> failed. The signal's manual inhibit attribute is set to false, not allowing writes.**

---

**Error Type:**

Error

**Possible Cause:**

The signal is configured not to allow manual changes during Runtime; as such, write operations are also not allowed.

**Solution:**

Check the device configuration's manual inhibit setting for the signal, then change it accordingly. Users may want to change the Server Tag's access to read only.

**Write to device <device name> signal <signal> failed. The signal is read only.**

---

**Error Type:**

Error

**Possible Cause:**

The driver attempted to write to a device signal that is read only.

**Solution:**

Check the device configuration to see if the signal is read only. If it is, then change the server tag to read only. This will cause the server to stop writes before they make it to the device.

**Write to device <device name> signal <signal> failed. The write value is invalid. Make sure the value is within the approved BSAP range.**

---

**Error Type:**

Error

**Possible Cause:**

1. The String, Float, or Boolean written to the device is malformed or is too large.
2. The type being written is different than the device data type (such as when writing a String to a Float signal).

**Solution:**

1. Ensure that the signal data type in the device matches the write.
2. If the signal is a String, ensure that the length does not exceed the device's 64-character limitation and is null terminated.

# Index

## <

- <Operation> on device <device name> signal <signal> caused an overflow. Some data has been lost. 17
- <Operation> on device <device name> signal <signal> failed. The device does not contain this signal. 18
- <Operation> on device <device name> signal <signal> failed. The device requires higher level permissions to perform this operation. 18
- <Operation> on device <device name> signal <signal> failed. The request was malformed or incomplete. 18

## A

- A read by address failed because the MSD version for device <device name> changed. The driver is sending read by name requests to get the updated MSD version from the device. 18
- A write by address failed because the MSD version for device <device name> changed. The driver is sending a write by name request to the device. 19
- Address Descriptions 13
- Automatic Tag Generation 14

## B

- BSAP Networks 5

## C

- Channel Setup 7
- Communication 8

## D

- Data Types Description 12
- Device Setup 8

## E

- Error Descriptions 17

## F

- Failed to send a request to device <device>. The device is not communicating. 19
- Failed to send a request to device <device>. The packet is malformed or too large. 19
- Failed to stop Winsock communications. 20

Failure to initiate winsock.dll [Error <error>]. 20

## I

Invalid BSAP function. 20

## O

Optimizing Communications 11

Overview 5

## R

Read failed on device <device> address <address>. The server expects this address to be of type <type>, but it is of type <type> in the device. 21

## T

Tag Import Settings 10

The response received from device <device> contains an invalid sequence number. Review timing properties. 21

The response received from device <device> is malformed. 21

The response received from device <device> is too large (%<count> bytes). The data is lost. 22

The signal was not found. 22

The string written to device <device>, address <address> is too long. The max supported string is <count> characters. The write will continue as is, but the device will truncate the value. 22

Timing 9

## U

Unable to bind to port <port> on Channel <channel>. 22

Unable to create a socket connection for Channel <channel>. 23

Unable to generate a tag database for device <device name>. Reason - Import file does not contain any valid signals. 23

## W

Write to device <device name> signal <signal> failed. The signal's manual inhibit attribute is set to false, not allowing writes. 23

Write to device <device name> signal <signal> failed. The signal is read only. 23

Write to device <device name> signal <signal> failed. The write value is invalid. Make sure the value is within the approved BSAP range. 24