



Irisys Java API for IRC3xxx Series (Blackfin-based) Devices

API Version 3.0.40518.01

Contents

1	Irisys Java API for IRC3xxx Series (Blackfin-based) Devices	4
1.1	Introduction	4
1.2	Release Notes	4
1.2.1	Version 3.0.40518.01 (May 2012)	4
1.2.2	Version 2.0.30810.01 (August 2011)	5
2	Basic Usage	7
2.1	Connecting to a Device	7
2.2	Using a Blackfin object to get device information	7
2.3	Disconnecting from a device	7
2.4	Setting IP board configuration	8
3	Application Notes	9
3.1	Using Irisys Timezones	9
3.1.1	Missing Timezones	9
3.2	Using Device Status for Troubleshooting	9
3.3	Packet Timeouts	10
4	Sample Applications	11
4.1	BlackfinConsole	11
4.2	BlackfinAPIServerSample	11
4.3	BlackfinAPIExampleGUI	11
5	Count Logs	12
5.1	Introduction	12
5.2	Logging Period	12
5.3	Accessing Count Logs	12
5.4	Storage	12
5.5	Resetting Count Logs	13
6	Frequently Asked Questions	14
7	Class Documentation	15
7.1	uk.co.irisys.Blackfin Class Reference	15
7.1.1	Detailed Description	16
7.1.2	Member Function Documentation	17
7.1.2.1	GetCurrentCount	17
7.1.2.2	GetDeviceStatus	17
7.1.2.3	GetUnitTime	17
7.1.2.4	GetUnitLocale	17
7.1.2.5	GetLocaleString	18
7.1.2.6	GetString	18
7.1.2.7	GetRegisterLabels	18
7.1.2.8	GetMACAddress	18
7.1.2.9	GetClientConfigIP	19
7.1.2.10	GetClientConfigHostname	19
7.1.2.11	GetClientConfigPort	19
7.1.2.12	GetClientConfigTimeout	19
7.1.2.13	GetClientConfigEnable	20

7.1.2.14	GetUnitTimeZone	20
7.1.2.15	GetDeviceName	20
7.1.2.16	GetDeviceID	21
7.1.2.17	GetSiteID	21
7.1.2.18	GetSerialNumber	21
7.1.2.19	GetCountLogInterval	21
7.1.2.20	GetCountLogPeriod	21
7.1.2.21	GetSiteName	22
7.1.2.22	GetUptime	22
7.1.2.23	GetUpTime	22
7.1.2.24	GetLastNCounts	22
7.1.2.25	GetCounts	23
7.1.2.26	GetDHCPEnabled	23
7.1.2.27	GetIPAddress	23
7.1.2.28	GetSubnetMask	24
7.1.2.29	GetGateway	24
7.1.2.30	GetIPFirmwareVersion	24
7.1.2.31	GetMonitorFirmwareVersion	24
7.1.2.32	GetDNS1	25
7.1.2.33	GetDNS2	25
7.1.2.34	GetDNS3	25
7.1.2.35	GetNetworkChecksum	25
7.1.2.36	SetCommsTimeout	26
7.1.2.37	SetPacketTimeout	26
7.1.2.38	PacketTimeout	26
7.1.2.39	SetUnitTime	26
7.1.2.40	SetUnitLocale	26
7.1.2.41	SetLocaleString	26
7.1.2.42	SetUserString	27
7.1.2.43	SetUnitTimeZone	27
7.1.2.44	SetDeviceName	27
7.1.2.45	SetDeviceID	28
7.1.2.46	SetSiteName	28
7.1.2.47	SetSiteID	28
7.1.2.48	SetCountLogPeriod	28
7.1.2.49	SetCountLogInterval	29
7.1.2.50	SetClientConfigIP	29
7.1.2.51	SetClientConfigHostname	29
7.1.2.52	SetClientConfigPort	30
7.1.2.53	SetClientConfigTimeout	30
7.1.2.54	SetClientConfigEnable	30
7.1.2.55	SetDNS1	31
7.1.2.56	SetDNS2	31
7.1.2.57	SetDNS3	31
7.1.2.58	ResetCountLogs	31
7.1.2.59	ResetCounts	32
7.1.2.60	ResetCurrentCount	32
7.1.2.61	ResetDeviceStatus	32
7.2	uk.co.irisys.BlackfinEngine Class Reference	32
7.2.1	Detailed Description	32
7.2.2	Member Function Documentation	33
7.2.2.1	GetAPIVersion	33

7.2.2.2	StartEngine	33
7.2.2.3	ShutdownEngine	33
7.2.2.4	AddNewCounterEndPoint	33
7.2.2.5	RemoveCounterEndPoint	33
7.3	uk.co.irisys.Blackfin.Count Class Reference	34
7.3.1	Member Data Documentation	34
7.3.1.1	countLines	34
7.3.1.2	countTime	34
7.4	uk.co.irisys.Blackfin.DeviceLogEntry Class Reference	34
7.4.1	Member Data Documentation	34
7.4.1.1	errorDescription	34
7.4.1.2	timestamp	34
7.5	uk.co.irisys.Blackfin.DeviceStatus Class Reference	35
7.5.1	Member Data Documentation	35
7.5.1.1	m_warnList	35
7.5.1.2	m_errorList	35
7.5.1.3	m_infoList	35
7.6	uk.co.irisys.IrisysTimeZone Class Reference	35
7.6.1	Constructor & Destructor Documentation	36
7.6.1.1	IrisysTimeZone	36
7.6.2	Member Function Documentation	36
7.6.2.1	setApplyDST	36
7.6.2.2	getApplyDST	36
7.7	uk.co.irisys.IrisysTimeZoneInfo Class Reference	36
7.7.1	Constructor & Destructor Documentation	37
7.7.1.1	IrisysTimeZoneInfo	37
7.7.2	Member Function Documentation	37
7.7.2.1	getSupportsDST	37
7.7.2.2	getBaseUTCOffset	37
7.7.2.3	getTimeZoneID	37

1 Irisys Java API for IRC3xxx Series (Blackfin-based) Devices

1.1 Introduction

The Irisys IRC3xxx series of People Counters provide TCP/IP connectivity via an IP master device. Irisys provides four Application Programming Interfaces (APIs) that developers can use to connect to and download count data from IRC3xxx series devices, in addition to setting and reading a number of user configurable identification strings on the device. The available APIs are;

- .NET (3.5)
- Java (1.6)
- Win32 (C++)
- Linux (C++)

Third parties can use these APIs to integrate count data from IRC3xxx series devices into their own custom applications. It is not intended to be used for the setup of counter networks or the retrieval of advanced data such as path maps or video, thus no functionality is provided for these tasks.

This documentation is intended for use by developers with a working knowledge of application programming and it is recommended to be read alongside the accompanying [Sample Applications](#). Note that this documentation assumes devices have been installed in accordance with our recommended practices, as outlined in the accompanying Installation Guide IPU40182 and Applications Guide IPU40184. Failure to follow these practices may affect the accuracy of the count data provided by IRC3xxx series devices.

The IRC3xxx series of devices use a Blackfin processing chip. Blackfin is a Registered Trademark of Analog Devices Inc.

1.2 Release Notes

1.2.1 Version 3.0.40518.01 (May 2012)

This release of the API adds support for new counter features in the latest firmware for IRC3xxx series devices, such as the ability to support up to 32 count registers.

Changelog

- Added new `uk.co.irisys.Blackfin.GetRegisterLabels()` function to download the new register labels
- The following methods can now throw an `IllegalArgumentException` when the provided string is too long to be stored on the device

- `uk.co.irisys.Blackfin.GetDeviceID()`
- `uk.co.irisys.Blackfin.GetDeviceName()`
- `uk.co.irisys.Blackfin.GetSiteID()`
- `uk.co.irisys.Blackfin.GetSiteName()`
- `uk.co.irisys.Blackfin.GetLocaleString()`

- `uk.co.irisys.Blackfin.GetUserString()`
- Removed unsafe `uk.co.irisys.BlackfinEngine.ClockCounters()` method which was deprecated by version 2.0 of this API
- Added new `uk.co.irisys.Blackfin.GetNetworkChecksum()` function a to create a checksum of the settings for all devices on the connected CAN network
- Fixed a data consistency issue when calling `uk.co.irisys.BlackfinEngine.AddNewCounterEndPoint` on an engine instance concurrently from multiple threads

1.2.2 Version 2.0.30810.01 (August 2011)

In this version of the API some functions have been renamed to provide consistency across all Irisys APIs for Blackfin-based devices. This is a breaking change from previous versions of the API and therefore any code based on an older version of the API may need to be updated to compile with this version. Other significant changes in this version include support for multiple (>2) count lines, Irisys time zones and improved documentation.

Changelog

- Added support for Irisys time zones
- Added support for getting & setting a new 'user string' setting
- All `uk.co.irisys.Blackfin.Set*` methods now have a `void` return type and throw the checked exception `IOException` upon failure.
- Methods returning reference types (such as `uk.co.irisys.Blackfin.GetDeviceName()`) will not now generate compiler warnings, but code to handle null reference return values is now unnecessary.
- The following methods now throw the unchecked exception `System.lang.IllegalArgumentException` upon receiving incorrect arguments, rather than the checked exception `System.lang.Exception`.

- `uk.co.irisys.Blackfin.SetCountLogInterval()`
- `uk.co.irisys.Blackfin.SetClientConfigPort()`
- `uk.co.irisys.Blackfin.SetClientConfigTimeout()`

- `IBlackfinCommsErrorHandler` is the new interface for detecting communications errors. This replaces the `BlackfinConnectionPoint`, `IBlackfinCommsError` interface.
- `uk.co.irisys.BlackfinEngine` no longer derives from `Thread`. The exposed methods should not have been used so this should require no code changes.
- An additional console style demo application is included
- Deprecated the following methods, each of which has been replaced with an equivalent method which is consistent with the other Blackfin APIs;
 - `uk.co.irisys.Blackfin.GetUnitLocale()`, which has been replaced with `uk.co.irisys.Blackfin.GetLocaleString()`
 - `uk.co.irisys.Blackfin.SetUnitLocale()`, which has been replaced with `uk.co.irisys.Blackfin.GetLocaleString()`
 - `uk.co.irisys.Blackfin.SetCommsTimeout()`, which has been replaced with `uk.co.irisys.Blackfin.SetPacketTimeout()`

- `uk.co.irisys.Blackfin.GetCountLogInterval()`, which has been replaced with `uk.co.irisys.Blackfin.GetCountLogPeriod()`
 - `uk.co.irisys.Blackfin.SetCountLogInterval()`, which has been replaced with `uk.co.irisys.Blackfin.SetCountLogPeriod()`
 - `uk.co.irisys.Blackfin.ResetCounts()`, which has been replaced with `uk.co.irisys.Blackfin.ResetCountLogs()`
 - `uk.co.irisys.Blackfin.GetUptime()`, which has been replaced with `uk.co.irisys.Blackfin.GetUpTime()`
- **Deprecated the redundant method `uk.co.irisys.BlackfinEngine.ClockCounters()`, which no longer serves any purpose and should not be called**
 - **Added class `uk.co.irisys.SerialNumberConverter` to convert numeric serial numbers to alpha numeric strings**

2 Basic Usage

This section provides an introduction to the basic usage of the Irisys API (Java) to connect to a device and perform some simple tasks with it. For more detailed information about the functions named in this section and other functions available in the API refer to the included class documentation.

2.1 Connecting to a Device

- Create a new object of type `uk.co.irisys.Blackfin`
- Establish a connection to the device:
 - Create a direct socket connection using `java.net.Socket` to the device IP address
 - Listen on an appropriate socket if the device has been configured for client connection mode, in which it will create an outgoing TCP/IP connection to a given IP address and port number. This can be configured via the web interface on the device or by using the People Counter Setup Tool
- After initializing the API, add the `uk.co.irisys.Blackfin` object to the `uk.co.irisys.BlackfinEngine` object by passing it with the `java.net.Socket` object to the `uk.co.irisys.BlackfinEngine.AddCounterEndPoint()` method
 - Optionally you can also specify an error handler so that you can be notified and take appropriate action if an error occurs during the connection process
- You can now use all methods provided by the `uk.co.irisys.Blackfin` object.

2.2 Using a Blackfin object to get device information

Once an `uk.co.irisys.Blackfin` object is connected to the counting network there are many methods which can be called on it. The getting and setting of the site name string is detailed below, but the general form for these methods is reflected in many other commands in the API.

- `string uk.co.irisys.Blackfin.GetSiteName()` throws `IOException`;
- `void uk.co.irisys.Blackfin.SetSiteName (String name)` throws `IOException`;

`uk.co.irisys.Blackfin.GetSiteName()` initiates a blocking communications call to the counter network, retrieving the site name string value from the master device and returning the retrieved string upon success or throwing an exception upon failure or timeout. Similarly `uk.co.irisys.Blackfin.SetSiteName()` sends the specified string to the master device which will be stored permanently.

2.3 Disconnecting from a device

- Remove the connected `uk.co.irisys.Blackfin` object from the `uk.co.irisys.BlackfinEngine` object by passing it to its `uk.co.irisys.BlackfinEngine.RemoveCounterEndPoint()` method. This disconnects the device from the socket or serial port.

2.4 Setting IP board configuration

Calling any of the `SetClientConfig*` or `SetDNS*` families of methods will cause the IP board on the device to be reset and your connection to the device will be lost. After calling any of these methods your `uk.co.irisys.Blackfin` object will be unable to communicate with the device and you should disconnect it and destroy the object before, if necessary, creating a new connection as outlined above. Note that the `GetClientConfig*` and `GetDNS*` families of methods are not affected by this.

It is recommended that these methods are only used when connected directly to the device, however they can also be used in client connection mode with the caveat that you may have to wait for a TCP/IP timeout to occur before you can communicate with the device again, which may take up to 10 minutes.

3 Application Notes

3.1 Using Irisys Timezones

As of API version 2.0.30810.01, it is now possible to get and set the time zone in which a device resides - the same setting which is available in the People Counter Setup Tool and Validation Tool. This setting does not affect the `uk.co.irisys.Blackfin.GetUnitTime()` function which always returns a value in UTC time.

The time zone setting is restricted to a (large) range of time zones which correspond to Windows time zones ID's. The `uk.co.irisys.IrisysTimeZoneInfo.getAllTimeZones()` method returns an array enumerating the set of time zones available. Each `uk.co.irisys.IrisysTimeZoneInfo` object has a base UTC offset and a Boolean value representing whether or not the time zone supports Daylight Savings Time (DST). To set the timezone a `uk.co.irisys.IrisysTimeZone` object should be constructed using the appropriate `uk.co.irisys.IrisysTimeZoneInfo` object for the desired timezone and `setApplyDST` should be called to specify whether or not DST is to be applied or not. Note that DST can only be applied where the `uk.co.irisys.IrisysTimeZoneInfo` specifies that it is supported.

If DST is to be taken into account when converting a UTC time to a local time then it is necessary to have more information than the API provides - the base UTC offset must be adjusted by a set of rules dependant on the particular time zone. These rules are non-trivial for many zones and are typically represented in a database. This information is available in the Windows registry, or by using the `System.TimeZoneInfo` class.

General procedure for calculating local time:

- Retrieve UTC time from device using `uk.co.irisys.Blackfin.GetUnitTime()`
- Retrieve time zone from device using `uk.co.irisys.Blackfin.GetUnitTimeZone()`
- If not applying daylight savings rules, add the time zone base UTC offset to the UTC time.
- If applying daylight savings rules:
 - Query database (e.g. the Windows Registry) for daylight savings rules
 - Apply rules to the UTC time to generate appropriate UTC offset for that point in time
 - Add UTC offset to the UTC time to get adjusted local time

3.1.1 Missing Timezones

Some of the time zone IDs contained within the Irisys API do not exist on some versions of Microsoft Windows when the latest time zone updates have not been installed, therefore you are advised not to assume the registry keys containing the time zone information exist on a given target machine.

You can add the missing time zones on Microsoft Windows NT 5.0 (XP, Server 2000) and newer by installing the latest time zone update via Windows Update or by downloading the latest time zone update package from the Microsoft Support website.

3.2 Using Device Status for Troubleshooting

The `uk.co.irisys.Blackfin.GetDeviceStatus()` method allows you to query the diagnostic log of the connected device. This can be used to determine if the device is in an error state or is issuing any warning messages, as well as informative messages such as the last reset time.

Note that IRC3xxx series devices will reset a device every minute in the event that a fatal error condition (such as array failure) has occurred. This ensures the device never becomes unresponsive, however it will cause any communications calls to be more likely to fail, especially those which take a long time to complete such as downloading the count log.

3.3 Packet Timeouts

Communications calls made by an `uk.co.irisys.Blackfin` instance are marked as having failed if no response is received within the configured timeout period, which defaults to 5 seconds. You can query the current timeout period using the `uk.co.irisys.Blackfin.PacketTimeout()` accessor function. For the majority of API functions this timeout is the maximum time the function will block for before returning a result, with the exceptions to this rule being;

- `uk.co.irisys.Blackfin.GetCounts()`
- `uk.co.irisys.Blackfin.GetLastNCounts()`
- `uk.co.irisys.Blackfin.GetDeviceStatus()`

If you are connecting to a device on a high latency network it may be useful to increase the timeout period using the `uk.co.irisys.Blackfin.SetPacketTimeout()` function to allow the API more time to wait for responses from that device, however this has the side effect of increasing the delay before a blocking API function will return a failure when the connection has been lost. One potential strategy is to count the number of failed API calls and, upon reaching a set threshold, raise the timeout value to try and work around network latency issues.

4 Sample Applications

This section provides a brief description of the sample applications included with the Java version of the API. If you have received this document without the sample applications listed below please contact your supplier.

The sample applications were developed using NetBeans 6.9.1 but should compile in most Java IDEs with little or no modification to the project files.

4.1 BlackfinConsole

A simple console application which demonstrates the correct procedure for connecting directly to a device using a TCP/IP socket and allows the user to test the various functions available through the API.

This simple application provides an easy way to experiment with the API and test the various methods available for interacting with IRC3000 series devices via the command line. It could easily be modified for use with batch or Powershell script files for automation of simple tasks.

The `BlackfinConsole.main` method creates and initialises a `uk.co.irisys.BlackfinEngine` object and then enters an infinite loop for user input. Upon the user entering a valid IP address it will attempt to create an outgoing socket connection to that IP address and, if successful, creates a `uk.co.irisys.Blackfin` object and associates it with the established socket by passing both to the engines `uk.co.irisys.BlackfinEngine.AddNewCounterEndPoint()` method.

Once a connection has been established the `BlackfinConsole.doConsoleCommandLoop` method is called, which enters another infinite loop for user input. The user can type in commands to execute and, if necessary, they will be prompted for argument values and the command will be executed. The help command can be used to list the available commands along with a short description and the disconnect command will break the infinite loop within this method.

Once execution returns from `BlackfinConsole.doConsoleCommandLoop` the `uk.co.irisys.Blackfin` object is passed to the engines `uk.co.irisys.BlackfinEngine.RemoveCounterEndPoint()` to disassociate it from the Socket and clean up resources and the infinite loop in the `BlackfinConsole.main` method iterates around to wait for a new IP address to connect to. Typing exit will quit the application.

4.2 BlackfinAPIServerSample

A simple console application which demonstrates how to listen for and accept client connections from IRC3xxx series devices and execute a set of API commands upon each device that connects.

4.3 BlackfinAPIExampleGUI

A GUI application demonstrating features of the API

This is a GUI program which provides a quick way to connect to IRC3xxx series devices and make changes to the settings using methods from the API. This application uses the swing toolkit to provide the interface.

The `ConnectHandler` class shows how to connect to a device.

The `initializeCommands()` method contains most of the `BlackfinAPI` related logic. This method adds event handlers to the buttons on the GUI which invoke the API commands and put the response back onto the GUI.

The `GetDeviceStatusCommand` class shows how to interpret the return value of `GetDeviceStatus`

5 Count Logs

5.1 Introduction

This section describes how Irisys IRC3xxx series devices handle count logs internally and how you can interact with them using the API. A count log is a record of the count values for each configured register at the time the log was written to the device's non-volatile memory.

5.2 Logging Period

The frequency with which count logs are created is determined by the `CountLogPeriod` setting which specifies, in seconds, the interval between each count log. The interval is based on the number of seconds since the start of the day to provide predictable logging times, hence the default logging interval of 900 seconds (15 minutes) will create count logs at 0000, 0015, 0030, etc.

You can use the `uk.co.irisys.Blackfin.GetCountLogPeriod()` API function to read the current value of this setting from the device. The `uk.co.irisys.Blackfin.SetCountLogPeriod()` API function or the People Counter Setup Tool can be used to change the value of this setting. Any changes to this setting will take effect immediately without any further action required.

5.3 Accessing Count Logs

There are two functions in the API for retrieving count logs from a device, `uk.co.irisys.Blackfin.GetCounts()` and `uk.co.irisys.Blackfin.GetLastNCounts()`. The first of these allows you to retrieve all count logs which fall into a specified time period, whilst the second retrieves a specified number of the most recent log entries.

Each of these functions returns an object of type `List<uk.co.irisys.Blackfin.Count>` containing an entry for each count log downloaded from the device. This class in turn contains a UTC timestamp representing the time the log entry was recorded and a `List<long>` object containing the count value for each configured countline. The size of this list depends on the number of registers configured at the time the log entry was recorded and may not be consistent between all of the log entries returned if the device configuration was changed during the log period retrieved.

5.4 Storage

Count logs are written to the device's non-volatile flash memory and therefore are not lost even if the power to the device is removed. The number of count log entries which can be stored on the device depends on the number of count registers which have been configured.

Due to the nature of the flash memory used on the device the only way to erase data is to erase the entire sector. Therefore the count logs are split between two sectors, with the oldest sector being erased when the newer sector is full. Assuming you do not reset the count logs at any point this means that the minimum number of stored count logs (after both flash sectors have been filled at least once) is half of the maximum theoretical capacity of the device and the number of counts available via the API will be any amount between the minimum and maximum.

The maximum and minimum storage capacities for a range of configured registers are given in the table below, along with the minimum number of days worth of logging this represents at 5, 15 and 60 minute logging intervals.

Registers	Minimum Capacity	Maximum Capacity	5 Minute Interval (Min)	15 Minute Interval (Min)	60 Minute Interval (Min)
2	4680	9361	16 Days	48 Days	195 Days
4	2978	5957	10 Days	31 Days	124 Days
8	1724	3448	5 Days	17 Days	71 Days
16	936	1872	3 Days	9 Days	39 Days
32	489	978	1 Day	5 Days	20 Days

The default logging period of 15 minutes with 2 enabled count registers will have a minimum capacity of 48 days worth of count log entries.

5.5 Resetting Count Logs

You can use the API to reset the count log on an IRC3xxx series device by calling the `uk.co.irisys.Blackfin.ResetCountLogs()` function, which will permanently erase all count log entries from the device. Note that flash memory has a finite lifespan and excessive use of this function could reduce the lifespan of your devices. Rather than resetting the count logs your application should keep track of the most recent count log entry retrieved from the device and use this as the starting point when requesting new data.

6 Frequently Asked Questions

What is the baud rate (speed) of an IP connection?

An Irisys IRC3xxx device can have a link speed of 10Mbps or 100Mbps, however the maximum data baud rate is limited to 1Mbps

Which TCP/IP ports are used to communicate with an Irisys IRC3xxx device?

Port 4505 is used to establish direct IP connections to the device.

Port 80 is used to connect to the web interface on the device and configure it using the built in People Counter Setup Tool (PCST) software.

By default port 5000 is used by devices for outgoing connections when client connection mode is enabled, however this can be configured via the web interface.

What is the default IP address of a device?

The factory default IP address is 192.168.0.10, using a subnet mask of 255.255.255.0

What is the maximum value of a count line?

The count registers on the device are 32 bits, giving a maximum value of over 4.2 billion, which is reset to 0 each time the device is powered off and on again.

What is the lifespan of the flash memory storage on the device?

The flash memory used on the device has a rated minimum lifespan of 100,000 erase / write cycles, which equates to roughly 26,000 years of logging at a 15 minute interval. Resetting the count log files will cause additional erase / write cycles to occur and will reduce the useful life of the device, therefore you should avoid doing this unless absolutely necessary. For more information about the count log files on the device see the [Count Logs](#) section.

7 Class Documentation

7.1 uk.co.irisys.Blackfin Class Reference

A representation of a IRC3000 series counting network This class represents an Irisys IRC3000 series counting unit (a master device) and optionally a set of node devices. It is used for all communication calls to the device including connection and disconnection.

Classes

- class [Count](#)

A single count log entry This structure contains the count log data for a single entry. It contains a list of count values corresponding to each count line from the device.

- class [DeviceLogEntry](#)

A single device status log entry This structure contains a string which describes a condition that the device was in at a specific time stamp.

- class [DeviceStatus](#)

A description of the current status of the device This structure contains a list of all of the errors, warnings and information messages for the device at the time of requesting.

Public Member Functions

- void [ResetCountLogs](#) () throws IOException
- void [ResetCounts](#) () throws IOException
- void [ResetCurrentCount](#) () throws IOException
- void [ResetDeviceStatus](#) () throws IOException

Setters

- boolean [SetCommsTimeout](#) (int milliseconds)
- void [SetPacketTimeout](#) (int milliseconds)
- int [PacketTimeout](#) ()
- void [SetUnitTime](#) (Calendar tm) throws IOException
- void [SetUnitLocale](#) (String locale) throws IOException
- void [SetLocaleString](#) (String locale) throws IOException
- void [SetUserString](#) (String userString) throws IOException
- void [SetUnitTimeZone](#) ([IrisysTimeZone](#) tz) throws IOException
- void [SetDeviceName](#) (String deviceName) throws IOException
- void [SetDeviceID](#) (String deviceID) throws IOException
- void [SetSiteName](#) (String siteName) throws IOException
- void [SetSiteID](#) (String siteID) throws IOException
- void [SetCountLogPeriod](#) (int countLogPeriod) throws IOException, IllegalArgumentException
- void [SetCountLogInterval](#) (Integer countLogInterval) throws IOException, IllegalArgumentException
- void [SetClientConfigIP](#) (Inet4Address ipAddress) throws IOException
- void [SetClientConfigHostname](#) (String hostname) throws IOException, IllegalArgumentException
- void [SetClientConfigPort](#) (Short portNumber) throws IOException, IllegalArgumentException
- void [SetClientConfigTimeout](#) (int timeout) throws IOException, IllegalArgumentException

- void [SetClientConfigEnable](#) (boolean enable) throws IOException
- void [SetDNS1](#) (Inet4Address ipAddress) throws IOException
- void [SetDNS2](#) (Inet4Address ipAddress) throws IOException
- void [SetDNS3](#) (Inet4Address ipAddress) throws IOException

Getters

- [Count GetCurrentCount](#) () throws IOException
- [DeviceStatus GetDeviceStatus](#) () throws IOException
- Calendar [GetUnitTime](#) () throws IOException
- String [GetUnitLocale](#) () throws IOException
- String [GetLocaleString](#) () throws IOException
- String [GetUserString](#) () throws IOException
- List< String > [GetRegisterLabels](#) () throws IOException
- String [GetMACAddress](#) () throws IOException
- Inet4Address [GetClientConfigIP](#) () throws IOException
- String [GetClientConfigHostname](#) () throws IOException
- short [GetClientConfigPort](#) () throws IOException
- int [GetClientConfigTimeout](#) () throws IOException
- boolean [GetClientConfigEnable](#) () throws IOException
- [IrisysTimeZone GetUnitTimeZone](#) () throws IOException
- String [GetDeviceName](#) () throws IOException
- String [GetDeviceID](#) () throws IOException
- String [GetSiteID](#) () throws IOException
- int [GetSerialNumber](#) () throws IOException
- int [GetCountLogInterval](#) () throws IOException
- int [GetCountLogPeriod](#) () throws IOException
- String [GetSiteName](#) () throws IOException
- BigInteger [GetUptime](#) () throws IOException
- BigInteger [GetUpTime](#) () throws IOException
- List< Count > [GetLastNCounts](#) (long n) throws IOException
- List< Count > [GetCounts](#) (Calendar startTime, Calendar endTime) throws IOException
- boolean [GetDHCPEnabled](#) () throws IOException
- Inet4Address [GetIPAddress](#) () throws IOException
- Inet4Address [GetSubnetMask](#) () throws IOException
- Inet4Address [GetGateway](#) () throws IOException
- String [GetIPFirmwareVersion](#) () throws IOException
- String [GetMonitorFirmwareVersion](#) () throws IOException
- Inet4Address [GetDNS1](#) () throws IOException
- Inet4Address [GetDNS2](#) () throws IOException
- Inet4Address [GetDNS3](#) () throws IOException
- String [GetNetworkChecksum](#) () throws IOException

7.1.1 Detailed Description

! class [Blackfin](#)

7.1.2 Member Function Documentation

7.1.2.1 Count `uk.co.irisys.Blackfin.GetCurrentCount ()` throws `IOException`

Get the live count line values from the device. These values are not written to the count log and therefore may not be the same as the most recent entry returned by the `GetCounts()` function

Returns

`Count` object containing the live values for each count line

Exceptions

<code>IOException</code>	If a comms error or timeout occurs
--------------------------	------------------------------------

7.1.2.2 DeviceStatus `uk.co.irisys.Blackfin.GetDeviceStatus ()` throws `IOException`

Get the device status logs since the last boot up or device log reset.

Note

This call may take longer than the configured packet timeout

Returns

`DeviceStatus` object containing the information, warning and error messages in the device status logs

Exceptions

<code>IOException</code>	If a comms error or timeout occurs
--------------------------	------------------------------------

7.1.2.3 Calendar `uk.co.irisys.Blackfin.GetUnitTime ()` throws `IOException`

Get the current UTC time of the devices internal clock, which is not affected by the configured timezone.

Returns

A `Calendar` object representing the devices UTC time

Exceptions

<code>IOException</code>	If a comms error or timeout occurs
--------------------------	------------------------------------

7.1.2.4 String `uk.co.irisys.Blackfin.GetUnitLocale ()` throws `IOException`

Returns

the locale string

7.1.2.5 String `uk.co.irisys.Blackfin.GetLocaleString ()` throws `IOException`

Get the locale string from the device

Returns

The retrieved locale string

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.6 String `uk.co.irisys.Blackfin.GetUserString ()` throws `IOException`

Get the user string from the device.

Returns

The retrieved user string

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.7 List<String> `uk.co.irisys.Blackfin.GetRegisterLabels ()` throws `IOException`

Retrieve the labels for the enabled count registers from the device.

This command was introduced in the 3.0.x series of APIs

Minimum DSP Firmware Version: 454

Returns

The retrieved count register labels

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.8 String `uk.co.irisys.Blackfin.GetMACAddress ()` throws `IOException`

Get the MAC address of the device, which must feature an IP board

Returns

A string representing the MAC address

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.9 Inet4Address uk.co.irisys.Blackfin.GetClientConfigIP () throws IOException

Get the client connection IP address configured on the device, which must feature an IP board. This setting specifies the IP address the device will attempt to connect to when client connection mode is enabled.

Note

If both an IP address and hostname are specified for client connection mode the IP address is used

Returns

The client connection IP address

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.10 String uk.co.irisys.Blackfin.GetClientConfigHostname () throws IOException

Get the client connection hostname configured on the device, which must feature an IP board. This setting specifies the hostname the device will attempt to connect to when client connection mode is enabled.

Note

If both an IP address and hostname are specified for client connection mode the IP address is used

Returns

The client connection host name

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.11 short uk.co.irisys.Blackfin.GetClientConfigPort () throws IOException

Get the client connection port configured on the device, which must feature an IP board. This setting specifies the port the device will attempt to connect to when client connection mode is enabled.

Returns

The client connection port number

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.12 int uk.co.irisys.Blackfin.GetClientConfigTimeout () throws IOException

Get the client connection timeout configured on the device, which must feature an IP board. This setting specifies the timeout, in seconds, between client connection attempts by the device when client connection mode is enabled.

Returns

The timeout period, in seconds

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.13 boolean uk.co.irisys.Blackfin.GetClientConfigEnable () throws IOException

Get the client connection state configured on the device, which must feature an IP board. This setting specifies whether the device will attempt to establish an outgoing connection to the IP address or hostname configured on the device.

Returns

The enabled state of client connection mode on the device

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.14 IrisysTimeZone uk.co.irisys.Blackfin.GetUnitTimeZone () throws IOException

Get the timezone configured on the device. Note that this has no functional impact on the device itself, which will always use UTC time.

Note

The timezone can be configured using the People Counter Setup Tool

Returns

An [IrisysTimeZone](#) object

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.15 String uk.co.irisys.Blackfin.GetDeviceName () throws IOException

Get the device name string from the device.

Returns

The retrieved device name string

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.16 String uk.co.irisys.Blackfin.GetDeviceID () throws IOException

Get the device ID string from the device.

Returns

The retrieved device ID string

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.17 String uk.co.irisys.Blackfin.GetSiteID () throws IOException

Get the site ID string from the device

Returns

The retrieved site ID string

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.18 int uk.co.irisys.Blackfin.GetSerialNumber () throws IOException

Get the unique serial number from the device

Returns

The unique serial number of the device

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.19 int uk.co.irisys.Blackfin.GetCountLogInterval () throws IOException

7.1.2.20 int uk.co.irisys.Blackfin.GetCountLogPeriod () throws IOException

Get the count log period currently configured on the device, which specifies the interval, in seconds, between count log entries being written to the devices memory.

Returns

The count log period in seconds

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.21 String uk.co.irisys.Blackfin.GetSiteName () throws IOException

Get the site name string from the device.

Returns

The retrieved site name string

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.22 BigInteger uk.co.irisys.Blackfin.GetUptime () throws IOException

7.1.2.23 BigInteger uk.co.irisys.Blackfin.GetUpTime () throws IOException

Get the elapsed time, in seconds, since the device was booted.

Returns

The devices current up time, in seconds

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.24 List<Count> uk.co.irisys.Blackfin.GetLastNCounts (long n) throws IOException

Get the latest N count log entries from the device.

Note

This call may take longer than the configured packet timeout
 The number of count log entries retrieved may be less than N if there are less than N entries in the device count log

Parameters

<i>n</i>	Number of count log entries to retrieve
----------	---

Returns

List of count log entries retrieved from the device

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.25 List<Count> uk.co.irisys.Blackfin.GetCounts (Calendar *startTime*, Calendar *endTime*) throws IOException

Gets the count log entries from the device which fall within the specified time range.

Note

This call may take longer than the configured packet timeout

Parameters

<i>startTime</i>	UTC time of the earliest count log entry to retrieve
<i>endTime</i>	UTC time of the latest count log entry to retrieve

Returns

List of count log entries retrieved from the device

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.26 boolean uk.co.irisys.Blackfin.GetDHCPEnabled () throws IOException

Get the DHCP state of the device, which must feature an IP board. This setting determines if the device obtains an IP address from a DHCP server or uses a statically configured address.

Returns

Current DHCP state of the device

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.27 InetAddress uk.co.irisys.Blackfin.GetIPAddress () throws IOException

Get the statically configured IP address of the device, which must feature an IP board. This setting determines the IP address of a device when it is not using DHCP

Returns

The static IP address configured on the device

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.28 Inet4Address uk.co.irisys.Blackfin.GetSubnetMask () throws IOException

Get the statically configured subnet mask of the device, which must feature an IP board. This setting determines the subnet mask of a device when it is not using DHCP.

Returns

The static subnet mask configured on the device

Exceptions

<i>IOException</i>	
--------------------	--

7.1.2.29 Inet4Address uk.co.irisys.Blackfin.GetGateway () throws IOException

Get the statically configured gateway of the device, which must feature an IP board. This setting determines the gateway of a device when it is not using DHCP.

Returns

The static gateway configured on the device

Exceptions

<i>IOException</i>	
--------------------	--

7.1.2.30 String uk.co.irisys.Blackfin.GetIPFirmwareVersion () throws IOException

Get the version of IP firmware installed on the device, which must feature an IP board

Returns

A string representing the IP firmware version

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.31 String uk.co.irisys.Blackfin.GetMonitorFirmwareVersion () throws IOException

Get the version of firmware installed on the device

Returns

A string representing the device firmware version

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.32 Inet4Address uk.co.irisys.Blackfin.GetDNS1 () throws IOException

Get the primary DNS server for the device, which must feature an IP board

Returns

An IP address

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.33 Inet4Address uk.co.irisys.Blackfin.GetDNS2 () throws IOException

Get the secondary DNS server for the device, which must feature an IP board

Returns

An IP address

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.34 Inet4Address uk.co.irisys.Blackfin.GetDNS3 () throws IOException

Get the tertiary DNS server for the device, which must feature an IP board

Returns

An IP address

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.35 String uk.co.irisys.Blackfin.GetNetworkChecksum () throws IOException

Generates a checksum of the current configuration settings of all devices on the CAN network of the connected device, which can be used to detect whether any configuration changes have taken place on the network between two subsequent calls to this function.

Note

This checksum may also be affected by firmware upgrades to any device on the CAN network

This command was introduced in the 3.0.x series of APIs

Exceptions

<i>IOException</i>	If a communications error occurs
--------------------	----------------------------------

Returns

7.1.2.36 `boolean uk.co.irisys.Blackfin.SetCommsTimeout (int milliseconds)`

7.1.2.37 `void uk.co.irisys.Blackfin.SetPacketTimeout (int milliseconds)`

Set the packet timeout value for the API to use when communicating with the connected device. See the notes on [Packet Timeouts](#) for more information.

Parameters

<i>milliseconds</i>	Timeout value, in milliseconds, to use during communications with the device
---------------------	--

7.1.2.38 `int uk.co.irisys.Blackfin.PacketTimeout ()`

Get the currently configured packet timeout value. See the notes on [Packet Timeouts](#) for more information.

Returns

Currently configured packet timeout in milliseconds

7.1.2.39 `void uk.co.irisys.Blackfin.SetUnitTime (Calendar tm) throws IOException`

Sets the internal clock on the device to the specified time, which should be in UTC

Parameters

<i>tm</i>	A Calendar object representing the UTC time to set on the device
-----------	--

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.40 `void uk.co.irisys.Blackfin.SetUnitLocale (String locale) throws IOException`

Parameters

<i>locale</i>	
---------------	--

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.41 `void uk.co.irisys.Blackfin.SetLocaleString (String locale) throws IOException`

Set the locale string on the device to the specified value

Parameters

<i>locale</i>	Locale string to set on the device
---------------	------------------------------------

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentException</i>	If the specified string exceeds the maximum permitted length

7.1.2.42 void uk.co.irisys.Blackfin.SetUserString (String *userString*) throws IOException

Set the user string on the device to the specified value

Parameters

<i>userString</i>	User string to set on the device
-------------------	----------------------------------

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentException</i>	If the specified string exceeds the maximum permitted length

7.1.2.43 void uk.co.irisys.Blackfin.SetUnitTimeZone (IrisysTimeZone *tz*) throws IOException

Set the time zone and DST state on the device. Note that this has no functional impact on the device itself, which will always use UTC time internally.

Parameters

<i>tz</i>	Irisys time zone to set on the device
-----------	---------------------------------------

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.44 void uk.co.irisys.Blackfin.SetDeviceName (String *deviceName*) throws IOException

Set the device name string on the device to the specified value

Parameters

<i>deviceName</i>	Device name string to set on the device
-------------------	---

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentException</i>	If the specified string exceeds the maximum permitted length

7.1.2.45 void uk.co.irisys.Blackfin.SetDeviceID (String deviceID) throws IOException

Set the device ID string on the device to the specified value

Parameters

<i>deviceID</i>	Device ID string to set on the device
-----------------	---------------------------------------

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentException</i>	If the specified string exceeds the maximum permitted length

7.1.2.46 void uk.co.irisys.Blackfin.SetSiteName (String siteName) throws IOException

Set the site name string on the device to the specified value

Parameters

<i>siteName</i>	Site name string to set on the device
-----------------	---------------------------------------

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentException</i>	If the specified string exceeds the maximum permitted length

7.1.2.47 void uk.co.irisys.Blackfin.SetSiteID (String siteID) throws IOException

Set the site ID string on the device to the specified value

Parameters

<i>siteID</i>	Site ID string to set on the device
---------------	-------------------------------------

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentException</i>	If the specified string exceeds the maximum permitted length

7.1.2.48 void uk.co.irisys.Blackfin.SetCountLogPeriod (int countLogPeriod) throws IOException, IllegalArgumentException

Set the count log period on the device which specifies the interval, in seconds, between count log entries being written to the devices memory.

Parameters

<i>countLogPeriod</i>	The count log period to set on the device, in seconds
-----------------------	---

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentExcep-tion</i>	if the period is not in the range 60 to 3600 inclusive.

7.1.2.49 void uk.co.irisys.Blackfin.SetCountLogInterval (Integer countLogInterval) throws IOException, IllegalArgumentExcep-tion

Parameters

<i>countLogInter-val</i>	Time in seconds between count log entries
--------------------------	---

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.50 void uk.co.irisys.Blackfin.SetClientConfigIP (Inet4Address ipAddress) throws IOException

Set the client connection IP address on the device, which must feature an IP board. This setting specifies the IP address the device will attempt to connect to when client connection mode is enabled. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Note

Set this value to 0.0.0.0 if you want to specify a hostname to connect to instead

Parameters

<i>ipAddress</i>	The client connection IP address to set on the device
------------------	---

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.51 void uk.co.irisys.Blackfin.SetClientConfigHostname (String hostname) throws IOException, IllegalArgumentExcep-tion

Set the client connection hostname on the device, which must feature an IP board. This setting specifies the hostname the device will attempt to connect to when client connection mode is enabled. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Note

If a client connection IP address other than 0.0.0.0 is specified this setting has no effect

Parameters

<i>hostname</i>	The client connection hostname to set on the device
-----------------	---

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.52 void uk.co.irisys.Blackfin.SetClientConfigPort (Short *portNumber*) throws IOException, IllegalArgumentException

Set the client connection port number on the device, which must feature an IP board. This setting specifies the port the device will attempt to connect to when client connection mode is enabled. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Parameters

<i>portNumber</i>	The client connection port number to set on the device
-------------------	--

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.53 void uk.co.irisys.Blackfin.SetClientConfigTimeout (int *timeout*) throws IOException, IllegalArgumentException

Set the client connection timeout on the device, which must feature an IP board. This setting specifies the timeout, in seconds, between client connection attempts by the device when client connection mode is enabled. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Parameters

<i>timeout</i>	The client connection timeout to set on the device, in seconds
----------------	--

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
<i>IllegalArgumentException</i>	If the timeout is < 1 or timeout > 172800

7.1.2.54 void uk.co.irisys.Blackfin.SetClientConfigEnable (boolean *enable*) throws IOException

Enable or disable client connection mode on the device, which must feature an IP board. If this setting is enabled the device will regularly attempt to establish an outgoing client connection to the IP address or hostname specified in the client connection settings. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Parameters

<i>enable</i>	Client connection enabled state to set on the device
---------------	--

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.55 void uk.co.irisys.Blackfin.SetDNS1 (Inet4Address ipAddress) throws IOException

Set the primary DNS server for the device to use when resolving hostnames, which must feature an IP board. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Parameters

<i>ipAddress</i>	Primary DNS server IP address to set on this device
------------------	---

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.56 void uk.co.irisys.Blackfin.SetDNS2 (Inet4Address ipAddress) throws IOException

Set the secondary DNS server for the device to use when resolving hostnames, which must feature an IP board. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Parameters

<i>ipAddress</i>	Secondary DNS server IP address to set on this device
------------------	---

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.57 void uk.co.irisys.Blackfin.SetDNS3 (Inet4Address ipAddress) throws IOException

Set the tertiary DNS server for the device to use when resolving hostnames, which must feature an IP board. Please refer to the notes on [Setting IP board configuration](#) before using this function with a TCP/IP connection.

Parameters

<i>ipAddress</i>	Tertiary DNS server IP address to set on this device
------------------	--

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.58 void uk.co.irisys.Blackfin.ResetCountLogs () throws IOException

Empty the count log stored on the device. This action cannot be undone and will not affect the live count line values.

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.59 void uk.co.irisys.Blackfin.ResetCounts () throws IOException

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.60 void uk.co.irisys.Blackfin.ResetCurrentCount () throws IOException

Reset the live count line values on the device. This action cannot be undone and does not affect existing count log entries stored on the device.

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.1.2.61 void uk.co.irisys.Blackfin.ResetDeviceStatus () throws IOException

Clear all entries in the device status logs. This action cannot be undone.

Exceptions

<i>IOException</i>	If a comms error or timeout occurs
--------------------	------------------------------------

7.2 uk.co.irisys.BlackfinEngine Class Reference

Connects `uk.co.irisys.Blackfin` objects to sockets.

Public Member Functions

- void `StartEngine` ()
- void `ShutdownEngine` ()
- void `AddNewCounterEndPoint` (final `Blackfin` counter, final `Socket` socket, final `IBlackfinCommsErrorHandler` commsErrorHandler) throws `IOException`
- `Socket` `RemoveCounterEndPoint` (`Blackfin` counter)

Static Public Member Functions

- static `String` `GetAPIVersion` ()

7.2.1 Detailed Description

This class allows `uk.co.irisys.Blackfin` to be registered with the API engine and attached to a `java.net.Socket` through which it should communicate with an IRC3xxx series device

7.2.2 Member Function Documentation

7.2.2.1 `static String uk.co.irisys.BlackfinEngine.GetAPIVersion () [static]`

Returns a string representing the executing version of the Irisys Blackfin API

Returns

A string containing the API version

7.2.2.2 `void uk.co.irisys.BlackfinEngine.StartEngine ()`

Initialises the API engine so that `uk.co.irisys.Blackfin` objects may be added and connected

7.2.2.3 `void uk.co.irisys.BlackfinEngine.ShutdownEngine ()`

Disconnects all devices, shuts down the API engine and releases resources. You should not call any methods on `uk.co.irisys.Blackfin` objects after calling this method.

7.2.2.4 `void uk.co.irisys.BlackfinEngine.AddNewCounterEndPoint (final Blackfin counter, final Socket socket, final IBlackfinCommsErrorHandler commsErrorHandler) throws IOException`

Registers a `uk.co.irisys.Blackfin` object with the API engine and attaches it to the provided `Socket` to begin communicating with the Irisys device to which it is connected

Parameters

<i>counter</i>	The <code>uk.co.irisys.Blackfin</code> object to connect
<i>socket</i>	The open socket to attach the <code>uk.co.irisys.Blackfin</code> object to
<i>commsErrorHandler</i>	Error handler object

Exceptions

<i>IOException</i>	On failure to connect (time out) or comms error
<i>IllegalArgumentException</i>	If any of the parameters are not valid
<i>IllegalStateException</i>	If the AP engine has not been started correctly

7.2.2.5 `Socket uk.co.irisys.BlackfinEngine.RemoveCounterEndPoint (Blackfin counter)`

Detaches the provided `uk.co.irisys.Blackfin` object from it's socket and removes it from the API engine. The {Socket} object to which it was previously attached is returned by this function, the caller is responsible for closing it and cleaning up any resources

Parameters

<i>counter</i>	The <code>uk.co.irisys.Blackfin</code> object to be removed
----------------	---

Returns

The {Socket} to which the `uk.co.irisys.Blackfin` object was attached

7.3 `uk.co.irisys.Blackfin.Count` Class Reference

A single count log entry This structure contains the count log data for a single entry. It contains a list of count values corresponding to each count line from the device.

Public Attributes

- List< Long > `countLines` = new ArrayList<Long>()
- Calendar `countTime`

7.3.1 Member Data Documentation

7.3.1.1 List<Long> `uk.co.irisys.Blackfin.Count.countLines` = new ArrayList<Long>()

A list of count values in line order (index 0 = count line 1, index 1 = count line 2)

7.3.1.2 Calendar `uk.co.irisys.Blackfin.Count.countTime`

The time at which the count was logged

7.4 `uk.co.irisys.Blackfin.DeviceLogEntry` Class Reference

A single device status log entry This structure contains a string which describes a condition that the device was in at a specific time stamp.

Public Attributes

- String `errorDescription` = ""
- Calendar `timestamp`

7.4.1 Member Data Documentation

7.4.1.1 String `uk.co.irisys.Blackfin.DeviceLogEntry.errorDescription` = ""

A string describing the state of the counter In the case of the messages log, this may not actually be an error

7.4.1.2 Calendar `uk.co.irisys.Blackfin.DeviceLogEntry.timestamp`

The timestamp at which the state was logged

7.5 uk.co.irisys.Blackfin.DeviceStatus Class Reference

A description of the current status of the device This structure contains a list of all of the errors, warnings and information messages for the device at the time of requesting.

Public Attributes

- List< DeviceLogEntry > `m_warnList` = new ArrayList<DeviceLogEntry>()
- List< DeviceLogEntry > `m_errorList` = new ArrayList<DeviceLogEntry>()
- List< DeviceLogEntry > `m_infoList` = new ArrayList<DeviceLogEntry>()

7.5.1 Member Data Documentation

7.5.1.1 List<DeviceLogEntry> uk.co.irisys.Blackfin.DeviceStatus.m_warnList = new ArrayList<DeviceLogEntry>()

A list of information messages logged on the device Currently this will only consist of the last reset time

7.5.1.2 List<DeviceLogEntry> uk.co.irisys.Blackfin.DeviceStatus.m_errorList = new ArrayList<DeviceLogEntry>()

A list of error messages logged on the device

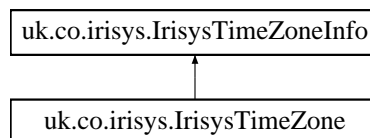
7.5.1.3 List<DeviceLogEntry> uk.co.irisys.Blackfin.DeviceStatus.m_infoList = new ArrayList<DeviceLogEntry>()

A list of warning messages logged on the device

7.6 uk.co.irisys.IrisysTimeZone Class Reference

Mutable class representing a time zone plus current settings for Adds a mutable "Apply Daylight Savings" field to the `IrisysTimeZoneInfo` immutable class.

Inheritance diagram for uk.co.irisys.IrisysTimeZone:



Public Member Functions

- void `setApplyDST` (boolean b)
- boolean `getApplyDST` ()
- `IrisysTimeZone` (`IrisysTimeZoneInfo` tz)

7.6.1 Constructor & Destructor Documentation

7.6.1.1 `uk.co.irisys.IrisysTimeZone.IrisysTimeZone (IrisysTimeZoneInfo tz)`

Create a new time zone from the specified time zone info object

Parameters

<i>tz</i>

7.6.2 Member Function Documentation

7.6.2.1 `void uk.co.irisys.IrisysTimeZone.setApplyDST (boolean b)`

Sets whether or not daylight savings values should be used to adjust local times

Parameters

<i>b</i>

7.6.2.2 `boolean uk.co.irisys.IrisysTimeZone.getApplyDST ()`

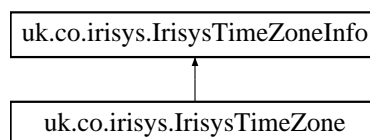
Returns a value determining whether or not Daylight Savings rules should be applied

Returns

7.7 `uk.co.irisys.IrisysTimeZoneInfo` Class Reference

Immutable class representing a time zone This class contains the primary field Timezone ID along with two useful pieces of information related to the ID- the base UTC offset and whether DST could be applied or not.

Inheritance diagram for `uk.co.irisys.IrisysTimeZoneInfo`:



Public Member Functions

- boolean `getSupportsDST ()`
- Date `getBaseUTCOffset ()`
- String `getTimeZoneID ()`

Protected Member Functions

- [IrisysTimeZoneInfo](#) (boolean dst, Date utc, String id)

7.7.1 Constructor & Destructor Documentation

7.7.1.1 `uk.co.irisys.IrisysTimeZoneInfo.IrisysTimeZoneInfo (boolean dst, Date utc, String id)` [protected]

Creates a new [IrisysTimeZoneInfo](#) object from the specified parameters

7.7.2 Member Function Documentation

7.7.2.1 `boolean uk.co.irisys.IrisysTimeZoneInfo.getSupportsDST ()`

Returns a value determining whether this time zone supports Daylight Savings

Returns

7.7.2.2 `Date uk.co.irisys.IrisysTimeZoneInfo.getBaseUTCOffset ()`

Returns the base UTC offset for this time zone (doesn't account for Daylight Savings)

Returns

A UTC offset

7.7.2.3 `String uk.co.irisys.IrisysTimeZoneInfo.getTimeZoneID ()`

Returns the corresponding Windows time zone ID

Returns