Irisys JavaAPI for IRC3xxx Series (Blackfin-based) Devices

API Version 3.0.50107.01

Contents

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 3.0.50107.01 (Jan 2013)

This is a minor re-release of version 3.0.41004.01

Changelog

- Fixed a race condition in the connection process
- Modified string bounds checking to use the byte length when encoded as UTF-8, rather than the number of characters

1.2.2 Version 3.0.41004.01 (Oct 2012)

This is a minor re-release of version 3.0.40518.01

Changelog

- Disabled some stderr output messages

1.2.3 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

1.2.4 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 recieving 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 chanages.
- An additional console style demo application is included
- Deprecated the following methods, each of which has been replaced with an equivilent 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 nofified 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 emunerating 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 no 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 Get-DeviceStatus

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

- **What is the baud rate (speed) of a serial connection?**

  The baud rate of a serial connection to an Irisys IRC3xxx device is 115200bps

- **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 I-RC3000 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, IllegalArgument-Exception
- void SetCountLogInterval (Integer countLogInterval) throws IOException, Illegal-ArgumentException
- void SetClientConfigIP (Inet4Address ipAddress) throws IOException
- void SetClientConfigHostname (String hostname) throws IOException, IllegalArgument-Exception
- void SetClientConfigPort (Short portNumber) throws IOException, IllegalArgument-Exception
- void SetClientConfigTimeout (int timeout) throws IOException, IllegalArgument-Exception
- 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

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

Returns

the locale string

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

Get the locale string from the device

Returns

The retrieved locale string

Exceptions

IOException	If a comms error or timeout occurs

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

Get the user string from the device.

Returns

The retrieved user string

Exceptions

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

Get the device name string from the device.

Returns

The retrieved device name string

Exceptions

IOException	If a comms error or timeout occurs

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

Get the device ID string from the device.

Returns

The retrieved device ID string

Exceptions

IOException	If a comms error or timeout occurs

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

Get the site ID string from the device

Returns

The retrieved site ID string

Exceptions

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

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

IOException	If a comms error or timeout occurs

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

Get the site name string from the device.

Returns

The retrieved site name string

Exceptions

IOException	If a comms error or timeout occurs

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

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

IOException	If a comms error or timeout occurs

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

n	Number of count log entries to retrieve

Returns

List of count log entries retrieved from the device

Exceptions

IOException	If a comms error or timeout occurs

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

startTime	UTC time of the earliest count log entry to retrieve
endTime	UTC time of the latest count log entry to retrieve

Returns

List of count log entries retrieved from the device

Exceptions

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

Inet4Address 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

IOException	If a comms error or timeout occurs

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

IOException	

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

IOException	

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

IOException	If a communications error occurs

Returns

boolean uk.co.irisys.Blackfin.SetCommsTimeout (int *milliseconds*)

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

milliseconds	Timeout value, in milliseconds, to use during communications with the device

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

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

tm	A Calender object representing the UTC time to set on the device

Exceptions

IOException	If a comms error or timeout occurs

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

Parameters

locale	

Exceptions

IOException	If a comms error or timeout occurs

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

Set the locale string on the device to the specified value
Parameters

locale	Locale string to set on the device

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	If the specified string exceeds the maximum permitted length

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

Set the user string on the device to the specified value
Parameters

userString	User string to set on the device

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	If the specified string exceeds the maximum permitted length

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

tz	Irisys time zone to set on the device

Exceptions

IOException	If a comms error or timeout occurs

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

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

deviceName	Device name string to set on the device

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	If the specified string exceeds the maximum permitted length

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

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

deviceID	Device ID string to set on the device

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	If the specified string exceeds the maximum permitted length

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

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

siteName	Site name string to set on the device

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	If the specified string exceeds the maximum permitted length

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

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

siteID	Site ID string to set on the device

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	If the specified string exceeds the maximum permitted length

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

countLog-Period	The count log period to set on the device, in seconds

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	if the period is not in the range 60 to 3600 inclusive.

void uk.co.irisys.Blackfin.SetCountLogInterval (Integer *countLogInterval*) throws IOException, IllegalArgumentException

Parameters

countLog-Interval	Time in seconds between count log entries

Exceptions

IOException	If a comms error or timeout occurs

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

ipAddress	The client connection IP address to set on the device

Exceptions

IOException	If a comms error or timeout occurs

void uk.co.irisys.Blackfin.SetClientConfigHostname (String *hostname*) throws IOException, IllegalArgumentException

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

hostname	The client connection hostname to set on the device

Exceptions

IOException	If a comms error or timeout occurs

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

portNumber	The client connection port number to set on the device

Exceptions

IOException	If a comms error or timeout occurs

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

timeout	The client connection timeout to set on the device, in seconds

Exceptions

IOException	If a comms error or timeout occurs
IllegalArgument-Exception	If the timeout is < 1 or timeout > 172800

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

enable	Client connection enabled state to set on the device

Exceptions

IOException	If a comms error or timeout occurs

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

ipAddress	Primary DNS server IP address to set on this device

Exceptions

IOException	If a comms error or timeout occurs

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

ipAddress	Secondary DNS server IP address to set on this device

Exceptions

IOException	If a comms error or timeout occurs

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

ipAddress	Tertiary DNS server IP address to set on this device

Exceptions

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

Exceptions

IOException	If a comms error or timeout occurs

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

IOException	If a comms error or timeout occurs

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

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

IOException	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 IBlackfin-CommsErrorHandler 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

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

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

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

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.

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

counter	The uk.co.irisys.Blackfin object to connect
socket	The open socket to attach the uk.co.irisys.Blackfin object to
commsError-Handler	Error handler object

Exceptions

IOException	On failure to connect (time out) or comms error
IllegalArgument-Exception	If any of the parameters are not valid
IllegalStateException	If the AP engine has not been started correctly

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 A-PI 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

counter	The uk.co.irisys.Blackfin 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

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)

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

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

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

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

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

A list of error messages logged on the device

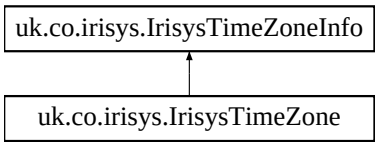
List<DeviceLogEntry> uk.co.irisys.Blackfin.DeviceStatus.m_infoList = new ArrayList<DeviceLogEntry>()

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)

Additional Inherited Members

7.6.1 Constructor & destructor documentation

uk.co.irisys.IrisysTimeZone.IrisysTimeZone (IrisysTimeZoneInfo *tz*)

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

Parameters

tz	

7.6.2 Member function documentation

void uk.co.irisys.IrisysTimeZone.setApplyDST (boolean *b*)

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

Parameters

b	

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

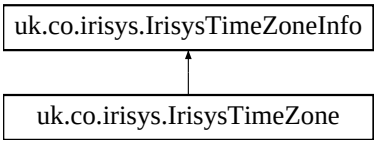
Returns a value determining whether or not Daylight Savings rules should be applied

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 ()

Static Public Member Functions

- static IrisysTimeZoneInfo[ ] getAllTimeZones ()

Protected Member Functions

- IrisysTimeZoneInfo (boolean dst, Date utc, String id)

7.7.1 Constructor & destructor documentation

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

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

Returns a value determining whether this time zone supports Daylight Savings

Returns

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

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

Returns the corresponding Windows time zone ID

Returns

static IrisysTimeZoneInfo [] uk.co.irisys.IrisysTimeZoneInfo.getAllTimeZones ( ) `[static]`

Returns a list of all available time zone objects

Returns

InfraRed Integrated Systems Limited

Park Circle Tithe Barn Way
Swan Valley
Northampton NN4 9BG UK
Tel: **+44 (0) 1604 594 200**
Fax: **+44 (0) 1604 594 210**
Email: **support@irisys.co.uk**
sales@irisys.co.uk

Web site: **www.irisys.co.uk**

Irisys Americas

One Glenlake Parkway
Suite 700
Atlanta GA 30328 USA
Tel: **+1 678 638 6248**
Email: **support@irisys.net**
sales@irisys.net

Web site: **www.irisys.net**

March 2014
IPU 40303
Issue 3