Warranties of individual software products are found in their respective manuals.
Table of Contents

PDF viewers: These page numbers refer to the printed version of this document. Use the PDF reader bookmarks tab for links to specific sections.

1. Introduction ................................................................. 1
   1.1 Background ................................................................................. 1
   1.2 Advantages of Using an SDK ....................................................... 1
   1.3 Overview of Current SDK Products ............................................. 2
       1.3.1 Current SDK Products ......................................................... 2
           1.3.1.1 LoggerNet SDK ........................................................ 2
           1.3.1.2 LNServer SDK ......................................................... 2
           1.3.1.3 BMP5 Direct SDK .................................................... 3
           1.3.1.4 Java PakBus SDK ..................................................... 3
       1.3.2 SDK Comparison Table ....................................................... 4

2. LoggerNet and LNServer SDK .................................... 4
   2.1 LoggerNet Overview .............................................................. 4
   2.2 LoggerNet and LNServer SDK Overview .................................. 5
       2.2.1 LoggerNet Server DLL ...................................................... 5
           2.2.1.1 Coralib3d.dll ......................................................... 5
           2.2.1.2 Coralib3.dll ........................................................ 5
       2.2.2 LoggerNet SDK Controls .................................................. 5
       2.2.3 LNServer SDK Controls ................................................... 6
       2.2.4 Registering the SDK Controls .......................................... 7
   2.3 Software Requirements ......................................................... 7
       2.3.1 Required Campbell Scientific, Inc. Software ......................... 7
       2.3.2 Development Tool Requirements ....................................... 7

3. BMP5 Direct SDK ..................................................... 7
   3.1 BMP5 Direct SDK Overview ..................................................... 7
   3.2 SimplePB.dll Wrapper ........................................................... 8

4. Java PakBus SDK ..................................................... 9
   4.1 Java PakBus SDK Overview ..................................................... 9
   4.2 Installing the Java PakBus SDK Components ............................ 9
       4.2.1 The /pakbus.jar File .................................................... 9
       4.2.2 The DOC Folder ........................................................ 9
       4.2.3 The SOURCE Folder .................................................... 9
   4.3 Development Tool Requirement ............................................. 10

Figure

1-1. Campbell Scientific SDK products functional overview .......... 2

Tables

2-1. LoggerNet SDK Controls and Uses ......................................... 6
2-2. LNServer SDK Controls and Uses ......................................... 6
1. Introduction

1.1 Background

Campbell Scientific dataloggers process measurements made with a wide variety of sensors and store summaries and statistics of these measurements as data. Dataloggers carry out these operations based on instructions in a datalogger program. Users create these datalogger programs with Campbell Scientific software such as Short Cut, Edlog, and CRBasic Editor. Programs can be sent to dataloggers via a variety of communication channels (for example, phone modem or RF).

A network of dataloggers may contain anywhere from one to several hundred dataloggers, each with its own set of sensors, communication links, and schedule for making measurements. Management of datalogger networks is an intricate task. Campbell Scientific software such as LoggerNet handles the task of network management by:

- Sending user-created programs to a datalogger.
- Checking computer and datalogger times and synchronizing if necessary.
- Managing the schedule of data collection from each datalogger.
- Storing values in the LoggerNet server data cache in tables that correspond to a given datalogger and datalogger program.
- Keeping log files for all communications, errors, etc.

Campbell Scientific customers have a wide variety of measurement interests—agriculture, weather, water resources, and vehicle testing, to name a few. Users often need applications that can be customized to suit their needs. The LoggerNet software product offers a number of general purpose clients that work with the LoggerNet server to perform the above tasks. However, there are times when users may need to write their own custom applications. Therefore, Campbell Scientific offers different Software Development Kits (SDK) that help developers create applications that extend or replace standard Campbell Scientific software products.

1.2 Advantages of Using an SDK

When developers need to write custom applications, SDK controls play an intermediary role (FIGURE 1-1) in performing desired tasks, such as sending a program to a specific datalogger or retrieving data from a datalogger. One can, in principle, achieve the same results without SDK controls by directly interacting with the LoggerNet server or the dataloggers through a "messaging" process, but this can be a very complex process. SDK controls greatly simplify
the communication process and provide a layer of insulation from future changes of the messaging protocol to the datalogger.

1.3 Overview of Current SDK Products

Campbell Scientific offers several different SDK products. Each SDK product provides different advantages depending on the specific needs of the developer.

1.3.1 Current SDK Products

1.3.1.1 LoggerNet SDK

This software development kit contains six ActiveX® controls and a limited version of the LoggerNet Server DLL. The included LoggerNet Server DLL (coralib3d.dll) can only be loaded and accessed locally and will only communicate with a single datalogger through a direct connection using RS-232 or an IP Port. However, a developer can also use this SDK to create applications that access an existing LoggerNet server installation.

The installation of this SDK includes a Programmer’s Reference and the source code for example applications written in Visual C++® (VS2013-MFC), C#, and VB.NET.

1.3.1.2 LNServer SDK

This software development kit contains six ActiveX controls and the unlimited version of the LoggerNet server. The included LoggerNet Server DLL (coralib3.dll) can only be loaded and accessed locally. However, a developer
can also use this SDK to create applications that access an existing LoggerNet server installation.

The installation of this SDK includes a *Programmer’s Reference* and the source code for example applications written in Visual C++ (VS2013-MFC), C#, and VB.NET.

### 1.3.1.3 BMP5 Direct SDK

This software development kit is for developers that want a simple way to create an application that communicates with any PakBus® datalogger using TCP/IP Sockets or a direct RS-232 link. Intermediate communication devices that are transparent to the connection method, such as RF401 radios, are permissible. The application created using the BMP5 Direct SDK will only be able to communicate with one datalogger at a time.

Starting with version 2.0, the BMP5 Direct SDK provides the developer access to a communications engine, coralib3d.dll, through the SimplePB.dll wrapper. The BMP5 Direct SDK install includes examples with source code along with documentation describing how to use the SimplePB.dll.

Previous versions of the BMP5 Direct SDK used the PakBusDLL.dll as the communication engine. The SimplePB.dll commands remain the same. Only the communication engine used by the SimplePB.dll has changed in version 2.0 of this SDK.

### 1.3.1.4 Java PakBus SDK

This software development kit is for developers wishing to target an application host other than the Microsoft® Windows® platform. Applicable only to PakBus dataloggers, this SDK requires no LoggerNet server for datalogger communication and supports concurrent dataloggers communications. The application will function as leaf node in the PakBus network; not a PakBus router. TCP/IP Sockets provide the simplest and most common method of datalogger communication with this SDK.
1.3.2 SDK Comparison Table

<table>
<thead>
<tr>
<th>SDK Name</th>
<th>Installed Items</th>
<th>Applications</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoggerNet SDK</td>
<td>• Limited LoggerNet Server (coralib3.dl) • Six ActiveX Control DLLs • Examples • Documentation</td>
<td>Best used for applications that will be connecting to an existing LoggerNet server or for standalone applications that don’t require concurrent or complex communication links. Works with all datalogger types.</td>
<td>Communicates with only a single datalogger at a time. Supports only direct TCP/IP or RS-232 connections.</td>
</tr>
<tr>
<td>LNServer SDK</td>
<td>• Full LoggerNet Server (coralib3.dll) • Six ActiveX Control DLLs • Examples • Documentation</td>
<td>Best used for standalone applications that require the robust communication capabilities and functionality of the full LoggerNet server. Works with all datalogger types.</td>
<td></td>
</tr>
<tr>
<td>BMP5 Direct SDK</td>
<td>• SimplePB.dll • Limited LoggerNet Server • Examples • Documentation</td>
<td>Best used for simple applications that communicate directly with PakBus dataloggers only.</td>
<td>Communicates with only a single PakBus datalogger at a time. Supports only direct TCP/IP or RS-232 connections. Limited functionality.</td>
</tr>
<tr>
<td>Java PakBus SDK</td>
<td>• JPakBus.jar • Documentation • Examples</td>
<td>Best used for standalone applications that are required to run on platforms other than Microsoft Windows.</td>
<td>PakBus dataloggers only. Not a PakBus router.</td>
</tr>
</tbody>
</table>

2. LoggerNet and LNServer SDK

2.1 LoggerNet Overview

*LoggerNet* is developed around client-server architecture. *LoggerNet’s* client-server technology is based on a server that communicates with a network of dataloggers via various communications technologies. The server listens for client requests, accepts the requests, and acknowledges to the client that a request has been received. The server fulfils this request and returns information to the client.

Often a client makes several requests. The server, however, processes only one request at time and in the order it was received. Once a client submits a request to the server, the client is free to do something else, knowing that its request will be processed. In other words, client requests may not get an immediate response from the server nor do the clients have to stop doing something else while waiting for an answer. This is called asynchronous communication.
2.2 LoggerNet and LNServer SDK Overview

The LoggerNet SDK and the LNServer SDK products differ only in the functionality of the LoggerNet Server DLL supplied with each, and the unique ActiveX® control required to start and stop the respective server. The remainder of the supplied components are common to each product and are identical in their function, operation, and use.

2.2.1 LoggerNet Server DLL

Fundamentally, the LoggerNet Server DLL provides the core functions of datalogger communication, data collection, and storage. Additionally, it functions as the ‘server’ component of a client-server architecture by exposing an API for client applications. It is this interface that is abstracted by the SDK.

The LoggerNet Server DLL does not need to be registered but must be placed in the application folder, in the PATH environmental variable, or in the Windows® system directory.

Depending on the product installed, one of two versions of the LoggerNet Server DLL is included with the SDK. The versions differ in the types of telecommunication devices supported.

2.2.1.1 Coralib3d.dll

The Coralib3d server is installed with the LoggerNet SDK. This limited function LoggerNet server supports only direct communications with the datalogger via RS-232, USB, or TCP/IP connections. The CsiServerDirect control is used to start and stop this server.

2.2.1.2 Coralib3.dll

The Coralib3 server is installed with the LNServer SDK. This LoggerNet Server DLL supports the same communications functionality available with the LoggerNet software. The CsiServer control is used to start and stop this server.

2.2.2 LoggerNet SDK Controls

The six ActiveX controls installed with the LoggerNet SDK are summarized in TABLE 2-1.
## TABLE 2-1. LoggerNet SDK Controls and Uses

<table>
<thead>
<tr>
<th>LoggerNet SDK Control</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>CsiBrokerMap.dll</td>
<td>Display names of dataloggers in the LoggerNet server’s network map</td>
</tr>
<tr>
<td></td>
<td>Display names of tables and columns of data from dataloggers</td>
</tr>
<tr>
<td>CsiDataLogger.dll</td>
<td>Establish a connection to a datalogger via the LoggerNet server</td>
</tr>
<tr>
<td></td>
<td>Send/Receive datalogger programs</td>
</tr>
<tr>
<td></td>
<td>Check datalogger time, synchronize time with PC</td>
</tr>
<tr>
<td></td>
<td>Retrieve data from a connected datalogger</td>
</tr>
<tr>
<td>CsiDataSource.dll</td>
<td>Monitor data collected from a datalogger</td>
</tr>
<tr>
<td>CsiCoraScript.dll</td>
<td>Execute CoraScript commands on the LoggerNet server</td>
</tr>
<tr>
<td>CsiLogMonitor.dll</td>
<td>Monitor LoggerNet server transaction and communication log files</td>
</tr>
<tr>
<td>CsiServerDirect.dll</td>
<td>Start and Stop the coralib3d.dll</td>
</tr>
</tbody>
</table>

### 2.2.3 LNServer SDK Controls

The six ActiveX controls installed with the LNServer SDK are summarized in TABLE 2-2.

## TABLE 2-2. LNServer SDK Controls and Uses

<table>
<thead>
<tr>
<th>LNServer SDK Control</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>CsiBrokerMap.dll</td>
<td>Display names of dataloggers in the LoggerNet server’s network map</td>
</tr>
<tr>
<td></td>
<td>Display names of tables and columns of data from dataloggers</td>
</tr>
<tr>
<td>CsiDataLogger.dll</td>
<td>Establish connections to dataloggers via the LoggerNet server</td>
</tr>
<tr>
<td></td>
<td>Send/Receive datalogger programs</td>
</tr>
<tr>
<td></td>
<td>Check datalogger time, synchronize time with PC</td>
</tr>
<tr>
<td></td>
<td>Retrieve data from a connected datalogger</td>
</tr>
<tr>
<td>CsiDataSource.dll</td>
<td>Monitor data collected from a datalogger</td>
</tr>
<tr>
<td>CsiCoraScript.dll</td>
<td>Execute CoraScript commands on the LoggerNet server</td>
</tr>
<tr>
<td>CsiLogMonitor.dll</td>
<td>Monitor LoggerNet server transaction and communication log messages</td>
</tr>
<tr>
<td>CsiServer.dll</td>
<td>Start and Stop the coralib3d.dll</td>
</tr>
</tbody>
</table>
2.2.4 Registering the SDK Controls

The ActiveX controls must be registered on the host PC before they can be imported into a development environment or accessed by a compiled application. The controls are registered on the development host by default during the installation of the SDK. Any controls used in the development of an application must also be registered on the application host. The latter can be accomplished as part of an installation program or done manually using the RegSvr32.exe utility installed with the Windows® OS.

For detailed information on importing the SDK controls into a Visual Studio® development environment, refer to the LoggerNet and LNServer SDK Programmer’s Reference.

2.3 Software Requirements

2.3.1 Required Campbell Scientific, Inc. Software

It is important to understand that it is the LoggerNet server that communicates directly with and collects data from a network of Campbell Scientific dataloggers. The SDK provides the means with which an application is able to ascertain and define the structure of the network, manage the server’s communications with the network, access the collected data, and monitor the server’s operation.

A version of the LoggerNet Server DLL is included and installed with the SDK. However, an application may also be created that utilizes an existing installation of LoggerNet software. The following Campbell Scientific software products provide the necessary LoggerNet Server DLL:

- LoggerNet version 1.1 or higher
- PC400 version 1.0 or higher
- VisualWeather 1.0 or higher

NOTE

The Allow Remote Connections option must be enabled on the remote LoggerNet Server DLL. (Remote: meaning not residing on the application host PC.)

2.3.2 Development Tool Requirements

The SDK ActiveX controls were developed in Visual C++ MFC environment. They are best suited for the Visual C++® MFC environment. However, the SDK controls can also be used in a .NET Framework environment. Examples are included for the Visual C++ (VS2013-MFC), C#, and VB.NET programming languages.

For details on developing a .NET application using the SDK ActiveX controls, refer to the LoggerNet and LNServer SDK Programmer’s Reference.

3. BMP5 Direct SDK

3.1 BMP5 Direct SDK Overview

The BMP5 Direct software development kit is for developers that want a simple way to create an application that communicates with any PakBus®
The BMP5 Direct SDK provides the developer with access to the communications engine, coralib3d.dll, through the SimplePB.dll wrapper. The BMP5 Direct SDK install includes examples with source code along with documentation describing how to use the SimplePB.dll.

### 3.2 SimplePB.dll Wrapper

The SimplePB.dll wrapper provides the developer with an easy interface through a communications engine to a single PakBus datalogger. The following basic commands are available in the SimplePB.dll:

- **OpenPort()** – used to open a COM port
- **ClosePort()** – used to close a COM port
- **OpenIPPort()** – used to open an IP port
- **CloseIPPort()** – used to close an IP port
- **GetClock()** – used to query a datalogger for its date and time
- **SetClock()** – used to set a datalogger’s date and time
- **GetValue()** – used to query a datalogger for a value or group of values
- **SetValue()** – used to set a field value in a datalogger
- **GetData()** – used to query data from a datalogger table
- **GetDataHeader()** – used to get just the header information for a specific table in the datalogger
- **GetCommaData()** – used to get just the CSV (comma-separated values) data from a specific table in the datalogger
- **File_Send()** – used to send a program file to a datalogger
- **GetAddress()** – used to query the PakBus address of a datalogger
- **GetStatus()** – used to query the current status of a datalogger
- **GetTableNames()** – used to query a datalogger for current table names and numbers
- **GetDLLVersion()** – used to return the version of SimplePB.dll currently in use
• GetLastResults() – used to display the results of a previous command that currently exist in memory as a string so memory pointers do not need to be managed

• FileControl – used to manage programs and files on the datalogger

• SetSecurity() – used to set the security code used to communicate with the datalogger

• GetTableRecordsCount() – used to determine the number of records that are available for collection from the specified table

4. Java PakBus SDK

4.1 Java PakBus SDK Overview

The Java PakBus® SDK is a simple API that can be used to write Java based applications that can communicate with Campbell Scientific dataloggers using PakBus protocol. An application built using this API should be able to function in any PakBus network and communicate with any datalogger in that network. The application will act as a PakBus leaf node. This means that it will not send or receive routing information with the exception of messages used to confirm neighbor links.

4.2 Installing the Java PakBus SDK Components

There is no install program required; simply copy the SDK components to a location on the development host. The SDK comprises a *jpakubs.jar* file, a *doc* folder, and a *source* folder.

4.2.1 The *jpakbus.jar* File

This Java archive file contains the compiled .CLASS files that will be accessed by the application at runtime. This file should be distributed with the application and placed in the application folder or have its location added to the CLASSPATH environment variable.

4.2.2 The *DOC* Folder

This folder contains the HTML formatted documentation for the SDK. To access the documentation, simply point a browser to the *index.html* file in the root of this folder.

As well as information and examples for using the SDK, the documentation provides detail descriptions of the packages, classes and interface that comprise the SDK. The *Java PakBus Software Development Kit* is an abridged version of this documentation and can be downloaded from the Campbell Scientific website.

4.2.3 The *SOURCE* Folder

This folder contains the source code files for the SDK examples as well as all of the classes and interfaces used by SDK. These files are included to aid in the understanding of the Java PakBus® SDK and cannot be redistributed, modified, or used as the basis for some other SDK/API product.
4.3 Development Tool Requirement

As a minimum, the *Java SE Software Development Kit 6* (JDK) or later will be required for application development. A comparable version of the *Java Virtual Machine* (JVM) must be installed on the application host.
Campbell Scientific Companies

Campbell Scientific, Inc.
815 West 1800 North
Logan, Utah  84321
UNITED STATES
www.campbellsci.com • info@campbellsci.com

Campbell Scientific Africa Pty. Ltd.
PO Box 2450
Somerset West 7129
SOUTH AFRICA
www.campbellsci.co.za • cleroux@csafrica.co.za

Campbell Scientific Southeast Asia Co., Ltd.
877/22 Nirvana@Work, Rama 9 Road
Suan Luang Subdistrict, Suan Luang District
Bangkok 10250
THAILAND
www.campbellsci.asia • info@campbellsci.asia

Campbell Scientific Australia Pty. Ltd.
PO Box 8108
Garbutt Post Shop QLD 4814
AUSTRALIA
www.campbellsci.com.au • info@campbellsci.com.au

Campbell Scientific (Beijing) Co., Ltd.
8B16, Floor 8 Tower B, Hanwei Plaza
7 Guanghua Road
Chaoyang, Beijing 100004
P.R. CHINA
www.campbellsci.com • info@campbellsci.com.cn

Campbell Scientific do Brasil Ltda.
Rua Apinagés, nbr. 2018 — Perdizes
CEP: 01258-00 — São Paulo — SP
BRASIL
www.campbellsci.com.br • vendas@campbellsci.com.br

Campbell Scientific Canada Corp.
14532 – 131 Avenue NW
Edmonton AB T5L 4X4
CANADA
www.campbellsci.ca • dataloggers@campbellsci.ca

Campbell Scientific Centro Caribe S.A.
300 N Cementerio, Edificio Breller
Santo Domingo, Heredia 40305
COSTA RICA
www.campbellsci.cc • info@campbellsci.cc

Campbell Scientific Ltd.
Campbell Park
80 Hathern Road
Shepshed, Loughborough LE12 9GX
UNITED KINGDOM
www.campbellsci.co.uk • sales@campbellsci.co.uk

Campbell Scientific Ltd.
3 Avenue de la Division Leclerc
92160 ANTONY
FRANCE
www.campbellsci.fr • info@campbellsci.fr

Campbell Scientific Ltd.
Fahrenheitstraße 13
28359 Bremen
GERMANY
www.campbellsci.de • info@campbellsci.de

Campbell Scientific Spain, S. L.
Avda. Pompeu Fabra 7-9, local 1
08024 Barcelona
SPAIN
www.campbellsci.es • info@campbellsci.es

Please visit www.campbellsci.com to obtain contact information for your local US or international representative.