



Product:	CDPDAQ
Product version:	V1.2
Document ID:	UM - CDPDAQ
Doc revision:	A
Date:	6. Nov. 2008
Pages:	14

Industrial Control Design AS



CDPDAQ V1.2 User Manual



Contents

1. INTRODUCTION.....	3
1.1. About.....	3
1.2. Component Overview.....	4
2. INSTALLATION AND CONFIGURATION.....	5
2.1. Installation.....	5
2.2. Configuration of CDPDAQ.....	5
2.2.1. Application.xml.....	5
2.2.2. CDPMail1.xml.....	6
2.2.3. DAQSignalLogger.xml.....	6
2.2.4. DAQIOServer.xml.....	6
2.2.5. Thruster1.xml.....	7
2.3. Configuration of DAQSimulator.....	8
2.3.1. Application.xml.....	8
2.3.2. DAQSim1.xml.....	8
2.4. Memory, CPU and network load considerations.....	9
2.5. Startup.....	9
3. CONTROLLING CDPDAQ.....	10
3.1. Logging frequency.....	10
3.2. Mail sending.....	11
4. FILE FORMAT.....	12
4.1. Index files.....	12
4.2. Mail index files.....	12
5. CONTROLLING DAQSIMULATOR.....	13
6. KNOWN ISSUES.....	14

1. Introduction

1.1. About

The application CDPDAQ was built with CDP components, providing the functionality to collect data discontinuously, and to store and to distribute it.

There are two different data sources:

- CDP signals from external CDP applications
- High frequency data delivered by CompactRIO microcontrollers

The application has been split in three components according to their functions:

1. DAQSignalLogger

- Receive data either from DAQIOServer or from CDPSignals
- Write data to logfiles
- Send message to mail component when mail sending of data is requested

2. DAQIOServer

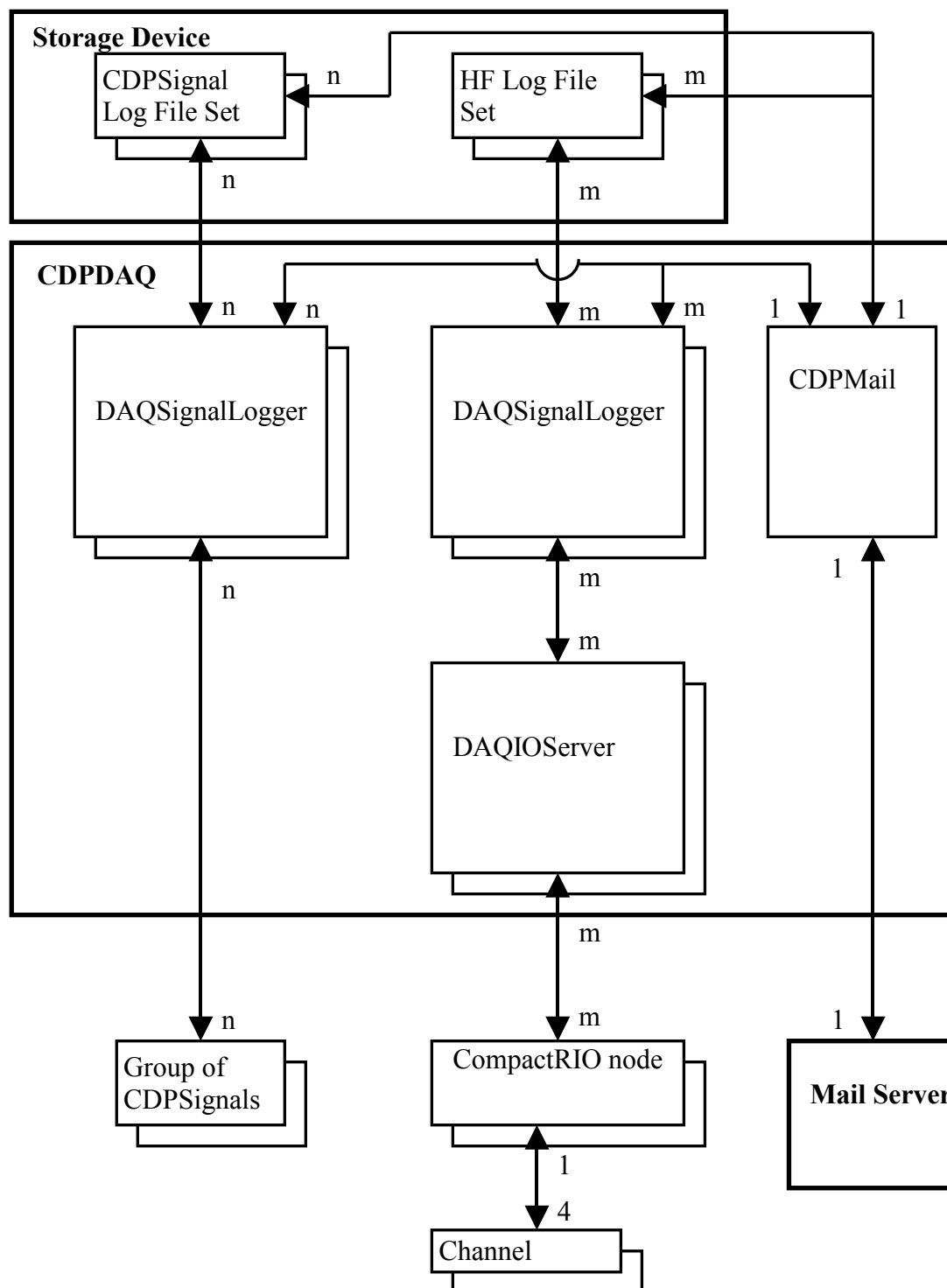
- Communicate with CompactRIO microcontroller
- Send measurement data to DAQSignalLogger

3. CDPMail

- ZIP data
- Send data as email using SMTP.

Additionally an application called DAQSimulator has been developed in order to simulate the behaviour of the CompactRIO measurement system.

1.2. Component Overview



2. Installation and Configuration

2.1. Installation

Unzip the zip file into a target directory. The following directory structure will come up:

```

Target\
  CDPDAQ\
    Application\
      configuration files for CDPSignal groups or HF data
      will be searched relative to this directory

      CDP.exe
      Application.xml
      Users.xml
    Components\
      Application\
        configuration of the instances of
        DAQIOServers and DAQSignalLoggers
        are expected in this directory

        CDPMail1.xml
        DAQSignalLogger1.xml
        DAQIOServer1.xml

      CDP\
        Models\
        WebServer\
  DAQSimulator\
    Application\
      CDP.exe
      Application.xml
      Users.xml
    Components\
      DAQSim1.xml
    CDP\
      Models\
      WebServer\
  
```

2.2. Configuration of CDPDAQ

2.2.1. Application.xml

Set a unique application name and handle for each instance of CDPDAQ.

```
<Application Name="CDPDAQ" Type="Application">
<Handle>6</Handle>
```

Make sure that exactly one instance of CDPMail is present.

```
<Subcomponent Name="CDPMail1" Model="CDPMail"></Subcomponent>
```

Add as many instances of DAQSignalLogger as individual Log Files are needed. Call the DAQSignalLoggers “DAQSignalLogger x ”, with x being a number.

```
<Subcomponent Name="DAQSignalLogger1"
Model="DAQSignalLogger"></Subcomponent>
```

Add as many instances of DAQIOServer as nodes to be logged at the CompactRIO controllers. Call them “DAQIOServer x ” with x being a number. Match this number with the corresponding DAQSignalLogger.

```
<Subcomponent Name="DAQIOServer1" Model="DAQIOServer"></Subcomponent>
```

2.2.2. CDPMail1.xml

Add the mail server properties:

```
<Mailserver>mx.company.com</Mailserver>
<Mailserverport>25</Mailserverport>
```

Add the credentials for SMTP AUTH authentication, if needed. If no authentication is needed, leave these values blank. In this case, consider other means to secure your mail server. Using no authentication is generally not recommended.

```
<Mailboxuser>User</Mailboxuser>
<Mailpassword>Password</Mailpassword>
```

Add the mail properties

```
<Mailfrom>sender@offshore.ship.com</Mailfrom>
<Mailto>receiver@headquarters.com</Mailto>
<Subject>CDPDAQ data mail with attachment</Subject>
<Mailbody>This is the email body text</Mailbody>
```

Set a maximum number of retries for mail sending. An alarm is raised, if there was a failure even on the last retry. Wait Retryinterval seconds between two retries.

```
<Retries>3</Retries>
<Retryinterval>20</Retryinterval>
```

Set the maximum number of lines in the Mail-Logfiles to the highest value of MaxNumberOfFiles of all DAQSignalLogger configuration files. These files are used to store the information, when a Log File has been processed and its result.

```
<MailindexLength>1000</MailindexLength>
```

You can activate the debug mode for mail sending by setting this value to 1. The complete communication between Mailserver and CDPMail will be printed to the program window.

```
<DEBUG>0</DEBUG>
```

2.2.3. DAQSignalLogger x .xml

Setup an individual DAQSignalLogger.xml file for each instance. Adjust the component name:

```
<Component Name="DAQSignalLogger1" Model="DAQSignalLogger">
```

Set the desired default and maximum logging frequency of the DAQSignalLogger. Take care to set this value reasonable high (i.e. at least to 1 Hz), as messages requesting a different logging frequency or the frequency controlling LogControl signal will be processed at this frequency.

```
<fs>10</fs>
```

Set the routing to a LogControl signal. This signal can be used to set the logging frequency and to start and stop logging. Set the Signal Name to “LogControl”, change the routing.

```
<Signal Name="LogControl" Input="1" Type="double" Unit=""
Routing="CDPLogManager.LogManager1.ControlSignal1" Description="Frequency
control signal from the LogManager"></Signal>
```

Add a connection to a DAQIOServer if the DAQSignalLogger shall be used to log HF data from a CompactRIO node.

```
<ComponentConnection Name="DAQIOServer1" Component="CDPDAQ.DAQIOServer1"></
ComponentConnection>
```

Add a connection to the CDPMail instance

```
<ComponentConnection Name="CDPMail"  
Component="CDPDAQ.CDPMail1"></ComponentConnection>
```

Add the filename to the configuration file. The path is relative to the “CDPDAQ\Application” directory

```
<ConfigFile>Thruster1.xml</ConfigFile>
```

2.2.4. DAQIOServerx.xml

Setup an individual DAQIOServer.xml file for each instance. Adjust the component name:

```
<Component Name="DAQIOServer1" Model="DAQIOServer">
```

Add a remote connection to the DAQSignalLogger that is used to log the HF signals

```
<ComponentConnection Name="DAQSignalLogger"  
Component="CDPDAQ.DAQSignalLogger1"></ComponentConnection>
```

Add the filename to the configuration file. The path is relative to the “CDPDAQ\Application” directory

```
<ConfigFile>Thruster1.xml</ConfigFile>
```

2.2.5. Thruster1.xml

Setup an individual configuration file for each instance of DAQSignalLogger. Match the filename with the appropriate ConfigFile value.

Take care to use '&' in the xml file, if the character '&' is to be printed in the Log File.

```
<Header lineprefix="% " filenameprefix="HF ">  
  <line name="Date" prefix="&amp;" value=""></line>  
  <line name="Time" prefix="&amp;" value=""></line>  
  <line name="Logging" prefix="" value="MANUAL"></line>  
  <line name="IMO Number" prefix="" value="IMO9181065"></line>  
  <line name="Unit Number" prefix="" value="xxxxnnnn"></line>  
  <line name="Signal Type" prefix="" value="VIBRATION"></line>  
  <line name="Measuring Time" prefix="" value="5"></line>  
  <line name="Sampling Frequency" prefix="" value="300"></line>  
  <line name="Azimuth" prefix="&amp;" value=""></line>  
  <line name="Pitch" prefix="&amp;" value="80"></line>  
  <line name="Load" prefix="&amp;" value=""></line>  
  <line name="Rpm" prefix="&amp;" value="1500"></line>  
</Header>  
<Cleartext>&amp;Acc01, &amp;Acc02, &amp;Acc03, &amp;ShaftDrivePositionRef</Cleartext>
```

Header lines will be setup as “Header lineprefix” + “line prefix” + “line name” + “=” + “value”.

Additionally a cleartext string can be defined, that is appended to the header without any prefixes.

The name of the Log File will be “filenameprefix” + “IMO Number” + “_” + “Unit Number” + “_” + Date (YYYYMMDD) + “_” + Time (HHMMSS) + FilenameSuffix. Date and Time correspond to the start time of logging.

Set the save directory for the Log Files

```
<SaveDirectory>c:\save</SaveDirectory>
```

Set the maximum number of files to keep on hard disk for this Log File Set. Old files will be deleted if this number is reached.

```
<MaxNumberOfFiles>1000</MaxNumberOfFiles>
```

Set a string that is used to separate data columns.

```
<Delimiterchar>,</Delimiterchar>
```

HF logging specific settings:

Set the number of measurement channels for the CompactRIO node
`<Measurementchannels>4</Measurementchannels>`

Set the CompactRIO controller properties, TCP port and IP:

`<CompactRIOport>27015</CompactRIOport>`

`<CompactRIOIP>127.0.0.1</CompactRIOIP>`

Set the CompactRIO node that shall be used for logging. Node numbering begins with 0.

`<Node>1</Node>`

Set the number of samples to be requested at a time. Reduce the value to reduce network load.

`<MaxNumberOfSamples>100</MaxNumberOfSamples>`

Set an additional delay in ms between two consecutive accesses to the CompactRIO controller. Increase this value to reduce network load.

`<TCPDelay>500</TCPDelay>`

CDPSignal logging specific settings:

Set a maximum file size for each Log File in kByte. A new Log File is used, if this size is exceeded.

`<MaxFileSize>500</MaxFileSize>`

Define the CDPSignals to be logged. A header line is generated out of this data and appended to the file header: "Prefix" + "Name" + "Delimiter Char" + "Prefix" + "Name" + ...

Take care to write '&' in xml if a '&' is desired in the Log File header.

```
<SignalGroup>
<SignalSource Prefix="&" Name="GeneratorSignal"
Routing="Generator.Generator.Output" Description="Signal1 is the first
signal."></SignalSource>
<SignalSource Prefix="&" Name="AnotherSignal" Routing="Thruster4.Temp"
Description="Signal2 is the second signal."></SignalSource>
</SignalGroup>
```

2.3. Configuration of DAQSimulator

2.3.1. Application.xml

Setup an individual Application name and Handle for each instance of the DAQSimulator application.

`<Application Name="Hostname_DAQSim1" Type="Application">`

`<Handle>9</Handle>`

2.3.2. DAQSim1.xml

Adjust the component name to an unique name:

`<Component Name="DAQSim1" Model="DAQSimulator">`

Set an individual free port for each instance of DAQSimulator, if they are run on the same machine.

`<LocalPort>27015</LocalPort>`

Set this value to the value of Node +1 of the corresponding DAQIOServer.

`<NumberOfNodes>8</NumberOfNodes>`

Set the number of channels to simulate.

`<NumberOfChannels>4</NumberOfChannels>`

This value is used to simulate a limited buffer size. Reply with **REQUEST_TOO_LARGE** error upon requests for $\text{Measuringtime} * \text{Samplingrate} > \text{MaxNumberofSamples}$.

```
<MaxNumberofSamples>1000000</MaxNumberofSamples>
```

2.4. Memory, CPU and network load considerations

The data transfer between the CompactRIO controller and the DAQIO Servers takes a considerable amount of time. Assuming 5 seconds of 4 channel measurement data with a measurement frequency of 50 kHz, 1 000 000 double values have to be transferred, corresponding to 64 million Bits. If this data is transferred in one single connection over a network bandwidth of 10 MBit/s, the corresponding DAQIO Server would block the network for 6.4 seconds. Therefore the `MaxNumberofSamples` parameter has to be used to reduce the network load and could be set to several hundreds. This will cause the DAQIO Server to transfer the data in several requests, allowing other applications in between to access the network.

Due to the split-up of the data transfer into several requests, a new client port is opened each time a new request is sent. By default, Windows XP reserves the ports 1025 - 5000 for client communication (see <http://www.microsoft.com/technet/community/columns/cableguy/cg1205.mspx>). As the ports need some time to close before they can be used again, the system can run out of free ports, if the time between data requests is too short. The `TCPDelay` parameter should therefore be set to values of about 1000 ms if several DAQIO Server instances are run at the same time.

Due to a high delay between consecutive data transfers and the small amount of data to be transferred in each connection, it can take around a quarter of an hour until all the data has been transferred. This cannot be circumvented if the network load has to be kept low. If network load is not a problem, the `MaxNumberofSamples` can be set to high values. In that case the `TCPDelay` parameter can also be decreased, thus leading to faster total data transfer.

2.5. Startup

The CDPDAQ application will be started by running the file:
`Installpath\CDPDAQ\Application\CDP.exe`

The DAQ Simulator is just used to simulate the CompactRIO measurement system. It will be started by running the file:
`Installpath\DAQSimulator\Application\CDP.exe`

3. Controlling CDPDAQ

The logging can either be controlled by the LogControl signal or by messages. If the parameter “Control signal active” is set to 1 for a given DAQSignalLogger, the LogControl signal is evaluated. Otherwise, control is achieved by messages.

3.1. Logging frequency

If “Control signal active” is set to 1 and the LogControl signal is greater than 0, logging is enabled. Additionally, the signal value is interpreted as desired logging frequency.

If “Control signal active” is not equal 1, logging will start on receipt of a LOGGING CDP text message and stop on receipt of a OFF CDP text message. The frequency can be controlled by sending a custom message, including the desired frequency. An appropriate code sample looks like this:

```
struct MessageDAQDouble
{
    Message header;
    double dParam;
};

double MessageFrequency;
CDPConnector DAQSignalLogger;

MessageDAQDouble m;
m.header.command=htonl(0x00000005);
m.header.origin=htonl(Handle());
m.dParam=htond(MessageFrequency);
m.header.parmaSize=htonl(sizeof(MessageDAQDouble)-sizeof(Message));
DAQSignalLogger->SendMessage((Message*)&m);
```

The logging frequency can be controlled by omitting a given number of steps when the ProcessLogging method is run. The logging frequency can only be set to values that are equal to integer fractions of the component frequency. Therefore the resulting frequency is higher, if a disallowed (intermediate) frequency has been requested.

Example:

Component frequency: 10 Hz

Allowed frequencies:

10 Hz / 1 = 10 Hz

10 Hz / 2 = 5 Hz

10 Hz / 3 = 3.3333 Hz

10 Hz / 4 = 2.5 Hz

10 Hz / 5 = 2 Hz

...

Requested frequency: 4 Hz

4 Hz is not an allowed frequency. Therefore the next higher, allowed frequency (5 Hz) is used.

3.2. Mail sending

If the parameter “SendFilesImmediately” is set to 1 for a given DAQSignalLogger, Log Files will be sent via e-mail as soon as they are ready for sending (e.g. Maximum file size reached for CDPSignal Log Files or all data has been written to a Log File for HF measurements).

Otherwise, Log Files will only be sent when requested. When sending a SENTALLDATA CDP text message to the DAQSignalLogger, all Log Files that are ready for sending and have not yet been sent without errors, will be sent via e-mail. When sending a SENTLASTDATA CDP text message to the DAQSignalLogger, the last Log File that is ready for sending and has not yet been sent without errors, will be sent via e-mail.

4. File format

4.1. Index files

Index files are used to keep track of the Log Files that are present in a Log File Set. They are named according to the Log Files “filenameprefix” + “IMO Number” + “_” + “Unit Number”.

There is an individual index file per Log File Set. Each line represents a filename that is present in the Log File Set.

To prevent the Log Files from filling the hard disk, only `MaxNumberOfFiles` lines will be stored in the index file. The oldest file will be removed from the hard disk if this number is exceeded.

```
CDP_IMO1234567_abcd1234
c:\temp\CDP_IMO1234567_abcd1234_20070620_104803.txt
c:\temp\CDP_IMO1234567_abcd1234_20070621_160303.txt
c:\temp\CDP_IMO1234567_abcd1234_20070621_171303.txt
```

4.2. Mail index files

Mail index files are used to keep track of the mail sending status for the files in a Log File Set. There is an individual mail index file per Log File Set. They are named according to the Log Files “filenameprefix” + “IMO Number” + “_” + “Unit Number” + “_mailindex.txt”.

Each line consists of the Date and Time of a mail sending event, the mail status and the name of the Log File that has been tried to send. It should be noted, that the mail status refers only to the mail delivery to the specified mail server and gives no information, if the mail has been received by the final recipient.

When a Log File is sent via email for another time, a previously present line for this Log File will be deleted from the mail index and a line containing the new mail sending status will be appended.

To prevent the mail index from becoming too large, only `MailindexLength` number of lines will be stored in the index. This parameter should be set to the largest `MaxNumberOfFiles` value of all `DAQSignalLogger` configurations to prevent sending of the same log file several times.

```
CDP_IMO1234567_abcd1234_mailindex.txt
2007/22/06 10:48:53:OK.:c:\temp\CDP_IMO1234567_abcd1234_20070620_104803.txt
2007/22/06 10:50:33:ERR:c:\temp\CDP_IMO1234567_abcd1234_20070621_160303.txt
```

5. Controlling DAQSimulator

Send a COMM text message to the DAQSimulator to enable communication for this instance.

The simulator will send double values to the DAQIOServer. The values will begin with 0 and will be increased by one, when all the channels have been transferred, i.e. for 4 data channels:

```
0 0 0 0  
1 1 1 1  
...  
N N N N
```

6. Known Issues

ICD Bug #	Description