RTP2EW

RTP Client

Rev 2.0.0

5/31/2013

This REF TEK manual describes an RTPD client program that is a data source module for Earthworm.

Revision History:

Revision	Date	Reason for change	Pages
2.0.0	**2013.06.05**	**Initial release**	**All**

Optional Manuals	PDF file
RTCC Command / Control Users Guide	RTCC.pdf
RT_Display Users Guide	RTDisplay.pdf
RT_View Users Guide	RTView.pdf
RTPMonitor Installation and Users Guide	RTPM.pdf
RTPD Installation and Users Guide	RTPD.pdf

Notation Conventions

The following notation conventions are used throughout REF TEK documentation:

Notation	Description
ASCII	Indicates the entry conforms to the American Standard Code for Information Interchange definition of character (text) information.
Binary	Indicates the entry is a raw, numeric value.
Hex	Indicates hexadecimal notation. This is used with both ASCII characters (0 – 9, A – F) and numeric values.
BCD	Indicates the entry is a numeric value where each four bits represents a decimal digit.
FPn	Indicates the entry is the ASCII representation of a floating-point number with n places following the decimal point.
<n>	Indicates a single 8-bit byte. When the contents are numeric, it indicates a hexadecimal numeric value; i.e. <84> represents hexadecimal 84 (132 decimal). When the contents are capital letters, it represents a named ASCII control character; i.e. <SP> represents a space character, <CR> represents a carriage return character and <LF> represents a line feed character.
MSB	Most Significant Byte of a multi-byte value.
MSbit	Most Significant Bit of a binary number.
LSB	Least Significant Byte of a multi-byte value.
LSbit	Least Significant Bit (bit 0) of a binary number.
YYYY	Year as a 4-digit number
DDD	Day of year
HH	Hour of day in 24-hour format
MM	Minutes of hour
SS	Seconds of minute
TTT	Thousandths of a second (milliseconds)
IIII	Unit ID number

n, nS	nano, nanoSecond; 10^{-9} = 0.000000001
u, uS	micro, microSecond; 10^{-6} = 0.000001
m, mS	milli, milliSecond; 10^{-3} = 0.001
K, KHz	Kilo, KiloHertz; 10^3 = 1,000
M, MHz	Mega, MegaHertz; 10^6 = 1,000,000
G, GHz	Giga, GigaHertz; 10^9 = 1,000,000,000
Kb, KB	Kilobit, KiloByte; 2^{10} = 1,024
Mb, MB	Megabit, MegaByte; 2^{20} = 1,048,576
Gb, GB	Gigabit, GigaByte; 2^{30} = 1,073,741,824

REF TEK   Support and update notifications

As a valued user of REF TEK equipment we would like to provide the best support possible by keeping you up to date with our product updates.

If you would like to be notified of any REF TEK product updates please spend a couple of minutes to register with the REF TEK customer support team.

To register, fill out our online registration form at http://support.reftek.com .

Once we register your contact we will only send necessary notifications via email. The same notifications will be shown on our website's http://support.reftek.com notifications page .

Thanks,

Your REF TEK support team

Contents

This page intentionally left blank.

1 General Description

RTP2EW (formerly known as REFTEK2EW) is an Earthworm module that is a client of RTPD, the REF TEK data server. It gets PASSCAL-format data from RTPD using the RTPD client protocol. The RTP2EW module is a classic "data source" Earthworm module, in that it:

- Has a "dot d" configuration file that follows the normal Earthworm syntax of configuration files,

- Connects to an Earthworm shared memory ring,

- Beats an alive heartbeat message,

- Sends status messages,

- Responds to Earthworm termination requests and

- Writes its data to an Earthworm ring in Tracebuf2 format.

RTP2EW requires a proper Earthworm configuration already running (outside the scope of this document).

Communication between a REF TEK data logger (Data Acquisition System - DAS) and the server machine is done via the REF TEK Protocol (RTP). RTP is designed to provide full-duplex, packet-oriented transport over UDP. RTPD is available for Windows, Linux, Mac OS X and Solaris. Refer to instructions provided by REF TEK for using RPTD on each platform.

RTP2EW is available for Windows, Linux, Mac OS X and Solaris.

This page intentionally left blank.

2   RTP2EW Configuration

To run RTP2EW in Earthworm, add the RTP2EW process to the startstop configuration file specified for the OS being used. (Note that every Earthworm module accepts the dot d file as the first and usually only argument in the command line; e.g. "`rtp2ew rtp2ew.d`" is how you set up the Process line in the **startstop** file). Then edit the **rtp2ew.d** file (refer to an example file below) to specify the two ring Ids to write its data to; *WaveRing* for Tracebuf2 packets and *RawRing* for the raw REF TEK PASSCAL packets.   The PASSCAL packets can be used for debugging or archiving purposes within Earthworm, but there are no standard modules that use these packets as is.   Edit the *MyModuleId* parameter to reflect the MOD_RTP2EW module Id, that should also be set in the ***earthworm.d*** parameter file. Then edit the *Server* parameter to reflect the IP and port of the RTPD server, and finally edit the **reftek.scnl** file to specify the DAS/stream/channel mapping to SCNL (Station Channel Network Location) names for identifying the Tracebuf2 packets (see example **reftek.scnl** file below).

If you monitor your Earthworm modules using statmgr, then set the **rtp2ew.desc** file as described in the example below and add this to the statmgr.d list of modules to monitor.

This page intentionally left blank.

3   Basic Earthworm Setup

3.1 Example rtp2ew.d Configuration File

An example rtp2ew.d file is shown below (comments are denoted by a #
symbol before the comment):

```
MyModuleId        MOD_RTP2EW        # module id for this process
WaveRing          WAVE_RING         # output ring for TRACEBUF messages
RawRing           RAW_RING          # output ring for REFTEK messages
HeartBeatInterval 30                # seconds between heartbeats
```

At least one WaveRing and RtpRing must be defined.
If you have selected a WaveRing to output TRACEBUF2 messages, then it is
necessary to define a mapping between the digitizer, stream, channel
information that is contained in each packet to the station, channel, net, and
pin number that is required by the TRACEBUF2 messages. There is no
standard way to do this in vanilla Earthworm, therefore import_rtp defines
yet another parameter file. If you don't specify this file, then import_rtp
issues a warning message on startup and then generates its own names.
Station name is the unit id, and channel name is built up from the stream
and channel values in the DT header (stream:chan), the network name is set
to "RT", and the pin number is set to -1.

```
SCNLFile    reftek.scnl
```

Give the fully qualified host name, or the IP address of the server in the
usual decimal dot notation, followed by the port number for the RTP service.
If you are running this on the same computer as the RTP server, then
specify "localhost" as the host name.

```
Server      localhost    7000
```

The API allows you to request data from a single, specific, DAS or from all
DAS(es). There is no provision in the API for requesting data from a subset
of the available digitizers. If you want data from a single DAS, give its unit
id here, otherwise enter 0 to select all digitizers.

```
DASid       0
```

You may filter the input based on packet type. The following types are supported: AD CD DS DT EH ET OM SC SH. List the ones you want, or enter ALL for all available.

```
PktMask      ET DT SC
```

You may filter the DT packets based on stream. List the stream id's you want to see. The stream codes here are the one-based values that are used by people, not the zero-based values that are in the headers.

PLEASE READ THIS SECTION BEFORE SETTING SendUnknownChan

The program has two modes of operation:
1. Track only the data for channels listed in the SCNLFile
2. Track all channels for which data is received.

Under mode 1, memory will be allocated to track the channels when the SCNL file is read. Once the channel list is read from the file, no further data allocation, or list sorting will be done, but data from all channels that are **NOT IN THE SCNL LIST**, will be dropped. Under mode 2, memory will be allocated to track any channels that are encountered by the module. Each time a new channel is encountered, it will be added to the list, and the list will be resorted. Channel list allocation is done in blocks, and it is possible that if a new channel was added to the input stream while the program was running, that it could require the program to:
- Allocate a new large block of memory
- Copy the list from one memory block to another
- Free an old block of memory
- Resort the list

all before any other packets could be processed. This would probably not be the end of the world, but does add another degree of volatility to the program. Mode 1 essentially requires the operator to predefine all of the desired channels in the SCNLList, but is very stable. Mode 2 is more flexible, but has more volatility.

```
SendUnknownChan   0
```

Set TimeJumpTolerance to the amount of clock slop to allow between the timestamps of data coming in, and the system clock of the machine on which this program runs. Time slop only applies to the amount that the system clock is behind the data time. tNow < tData

```
Default: TimeJumpTolerance 86400 sec (one day).
```

The program will not write data packets whose timestamps are greater than tNow + TimeJumpTolerance (seconds)

```
TimeJumpTolerance 86400
```

```
DropPacketsOutOfOrder
```
Set a flag to control whether out-of-chronological-order packets will be ignored or not. Standard Earthworm modules may misbehave if they see out-of-order data (especially wave_serverV). Winston waveserver can handle out-of-order data.

```
Default: DropPacketsOutOfOrder 1
DropPacketsOutOfOrder 1
```

Set a flag to control whether packets with decompression errors will be dropped (not sent).

```
Default: DropPacketsWithDecompressionErrors 1
DropPacketsWithDecompressionErrors 1
```

3.2 Acceptable Sample Rates

The program will attempt to determine the nominal sample rate for each channel. The channel rate will only be set if it matches a valid sample rate as specified in the config file. So if rpt2ew starts up, and the first 3 packets it sees from a channel have respective rates of 125.0, 40.0, 40.0 it will set the sample rate to 40.0.

```
Default: AcceptableSampleRates 40.0 80.0 100.0 200.0
AcceptableSampleRates 40 80 100 200 250 500
```

SendTimeTearPackets
1

```
Default: SendTimeTearPackets 1
SendTimeTearPackets 1
```

3.3 Example of rtp2ew.desc Descriptor File

```
modName     rtp2ew
modId       MOD_RTP2EW
instId      INST_MENLO

tsec: 60  page: 0  mail: 10
```

Uncomment the `"restartMe"` line to enable automatic restart of this process by statmgr/startstop. statmgr will issue a `TYPE_RESTART` message for this process_id if it declares the patient dead

```
err: 1  nerr: 1  tsec: 0  page: 0  mail: 20
text: "RTP server OK."

err: 2  nerr: 1  tsec: 0  page: 0  mail: 20
text: "RTP server not responding."

err: 3  nerr: 1  tsec: 0  page: 0  mail: 20
text: "no data from RTP server."
```

3.4 An example reftek.scnl file

This file defines the mapping between the unit, stream, and channel entries which are in the DT packet headers and the station, channel, net, location, pin number parameters which are required by the TracePacket format. This file is referred to by the **rtp2ew.d** file by the SCNLFile argument.

Note: Stream number here is the one-based value that is used by people, not the zero-based value that is actually in the headers. The format of the reftek.scnl file is shown below in a table for clarity, but any white space characters should be used between each field.

DAS			Trace Packet					
#Unit ID	Stream	Chan	Station	Chan	Net	Loc	Pinno	Samp Rate
A902	1	1	PS1	EHZ	XR	00	0	100
A902	1	2	PS1	EHN	XR	00	00	100
A902	1	3	PS1	EHE	XR	00	0	100

This page intentionally left blank.

Trimble Navigation Limited
1600 Tenth Street
Suite A
Plano, Texas 75074
Tel: 214-440-1265
Fax: 972-578-0045
www.reftek.com

REF TEK

A DIVISION OF TRIMBLE