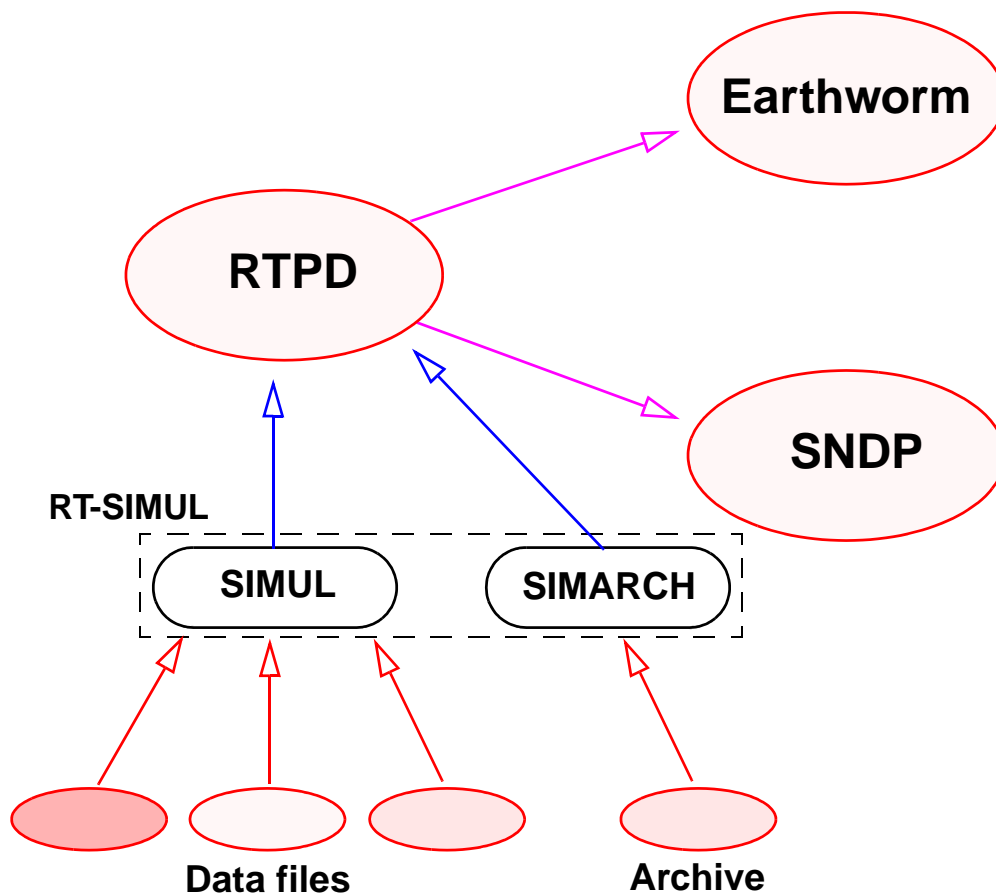


RT-SIMUL Software Users and Installation Guide

Refraction Technology



Refraction Technology
2626 Lombardy Lane
Dallas, Texas 75220
USA

Telephone: (214) 353-0609
FAX: (214)353-9659
EMAIL: info@reftek.com or support@reftek.com
FTP: <ftp.reftek.com>
WWW: <http://reftek.com>

Copyright 2001 Refraction Technology, Inc.

All rights reserved.

Printed in USA

This document is located at: M:\Archiver Utilities\RTSIM\RTSIMUL-S-003-A.book

Who Should Use This Manual

This guide was written assuming the user has basic familiarity with **RTPD** and REF TEK Utilities as well as an understanding of REF TEK Archive Utilities and its organization.

Purpose

This manual comprises two modules of the **RT-SIMUL** simulation software called **SIMUL** and **SIMARCH**.

The **SIMUL** software was designed to provide data processing modeling using pre-recorded data playback operations. **SIMUL (RS)** reads data from files and creates streams of REF TEK data packets to **RTPD**. **RTPD** accepts these data streams as normal data streams coming from REF TEK digitizers and passes this data to its client port. As a result, any application having an interface to **RTPD** can utilize and process this data from pre-recorded data files that are like real-time data from field stations.. Therefore **RS** is a REF TEK Network software model. So **RS** may be used for tuning **Earthworm** or **SNDP** real-time data processing systems for better seismic event location or event detection. **RS** has a distinctive feature in its ability to accelerate data transfer rate.

The **SIMARCH** software program is intended for reading data directly from REF TEK archive organized by **RTPD** and simulating real-time data streams to the designated **RTPD** (only **DT** packets are transferred where one hour of data (recorded during one hour) may be played back and re-processed within several minutes. This program differs from the **SIMUL** program. **SIMUL** reads files fetched from the archive. **SIMARCH** does not require file extraction from the archive.

Revision History

Revision	Date	ECN	Reason for change
A	09/25/01		Initial issue

Related Manuals:

REF TEK Utilities
RTPD Installation and User Guide
SNDP Reference Guide
Earthworm

Section: 1	<i>SIMUL REF TEK simulation software</i>	1
	General description - - - - -	1
	Model description - - - - -	2
	Control process simctrl	2
	Data sample sequence simulation.	4
	Simple data sequence generation	5
 Section: 2	 <i>SIMARCH – REF TEK real-time data stream simulation program</i>	 6
	Command format - - - - -	6
	Configuration file example: - - - - -	7
	Description of parameters: - - - - -	8
	Algorithm of work - - - - -	9
 Section: 3	 <i>Installation and Operation</i>	 10
	Introduction - - - - -	10
	Configure the application - - - - -	10
	Start the application - - - - -	10
	Diagnostics to check data path - - - - -	11
	Getting started - - - - -	12
	Description of messages	13
	Example of the configuration file:	14

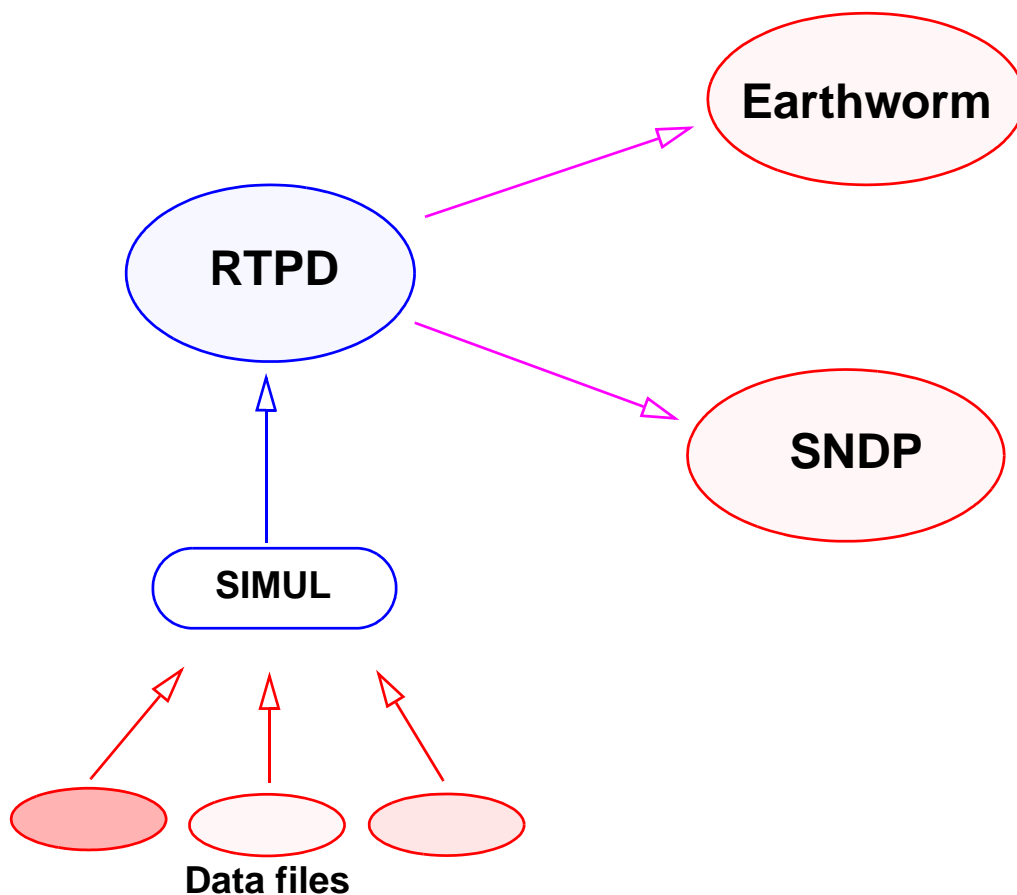
Section: 1 SIMUL REF TEK simulation software

1.1 • General description

SIMUL software was designed to provide data processing modeling using pre-recorded data playback operations. **SIMUL (RS)** reads data from files and creates streams of REF TEK data packets to **RTPD**. **RTPD** accepts these data streams as normal data streams coming from REF TEK digitizers and passes this data to its client port. As a result, any application having an interface to **RTPD** can utilize and process this data from pre-recorded data files that are like real-time data from field stations.. Therefore **RS** is a REF TEK Network software model.

So **RS** may be used for tuning **Earthworm** or **SNDP** real-time data processing systems for better seismic event location or event detection. **RS** has a distinctive feature in its ability to accelerate data transfer rate. This feature is better implemented in **SIMARCH** program, where one hour of data (recorded during one hour) may be played back and re-processed within several minutes.

RS is similar to **SIMARCH** program. However it reads data from the files previously extracted from the archive, and not from the entire archive.



1.2 • Model description

The model consists of the control process **simctrl** and a DAS - simulating processes. These simulating processes are:

Process	Use
simref	For REF TEK files (prepared by an arcfetch procedure from REF TEK archive)
simcon	For non packetized data (just samples)
simsim	Samples generating.

1.2.1 Control process simctrl

1. Process **simctrl** reads a configuration file and launches DAS emulators (**simref**).
2. It passes them the names of the configuration (**config**) files (one unique config file per each **simref**), RTPD host address, and port number. If the **PARM** field is set, then the fields **Delay**, **Start**, **Type** and **Debug** become active and the corresponding parameters are also passed to **simref**.
3. Process **simctrl** controls the termination of the child processes and displays reports on their termination. When terminating **simctrl** (SIGINT or SIGTERM UNIX signals), child processes are also terminated (kill SIGTERM).
4. If field **TYPE** is set to 1 or 3 (**Type**=1, or **Type**=3), then after all emulators stop (end of data) they are restarted again automatically (**Type**=3) or by prompt request (**Type**=1).

Note: Process **simctrl** can read arguments or search for master configuration file **simctrl.ini**

Master configuration file structure.

```
# RTPD server description:
# RTPD = AddrTo [, PortTo]
# AddrTo = host_name | IP address
# PortTo = port_number
# If RTPD is not defined, then AddrTo=localhost, PortTo=RTP_DEFAULT_PORT (2543)
# If PortTo is not defined, then PortTo = RTP_DEFAULT_PORT (2543)
RTPD=192.168.100.75
# Description of parameters common for all DAS(es): Delay, Start, Type, Debug:
# PARM = Delay, Start, Type, Debug
# If PARM is on place, then all DAS(es) will use same parameters, and
# parameters in self config files are ignored.
# If Type=3 all DAS(es) start over after end of data
# Notation "PARM=" is equivalent to "PARM=0,1,0,1
# Example for Delay=1, Start=-1, Type=0, Debug=3:
PARM = 1,-1,-0,3
# Description of DASA(es) to be simulated:
# DAS = module_name, config_file_name [, ini]
# module_name - {simref | simcon | simsim}
#simref - for REF TEK data files (see 2.2.1)
#simcon - for files with data samples (see 2.2.2)
#simsim - for generated samples (see 2.2.3)
# config_file_name - particular DAS configuration file
# ini - {0 | 1} - common parameters setting
# If ini=0, or not set, then DAS uses common parameters set in PARM
# If ini = 1, then parameters from particular config file is used
# In the following examples for DAS Unit ID=8021 self parameters will be used
# except for cases with Type = 3),
# and for Unit ID 8022 and 8023 - common parameters set in PARM
DAS = simref, 8021.ini,1
DAS = simcon, 8022.ini
DAS = simsim, 8023.ini
```

Emulator process.

The DAS-emulator processes receives a **config** file name and **RTPD**'s address parameters during launch.

REF TEK packets simulation.

1. Process **simref** can simulate work of one or more DAS units by utilizing REF TEK data files acquired by the **arcfetch** procedure from a REF TEK archive.
2. The REF TEK archive in its turn is created by **RTPD**.
3. The process connects to **RTPD** with an assigned in config file address and sends the data from the assigned files to its UDP-port.

Note: A user can run several **RTPD** instances with previously assigned different UDP and client ports in *rtpd.ini* files. So several networks at a time might be simulated at one host.

Parameters description.

The parameters may be passed to **simref** at initial startup or in config files. These are:

- Unit ID
- Packet delivery speed, PDS (same as recorded (real), or not)
- Time shift
- Playback mode.

Parameter	Description
Change of the Unit ID	Allows use of the same config file for different DAS simulations.
PDS	Maybe set to real, or with predefined time intervals. In the first case data packets will be delivered to RTPD according to the real delays (the delays are defined according to the time difference of neighboring packets in source files rounded to seconds). In the second case the in-between data packets interval is determined by the fixed time (in seconds).
Time shift	Allows the user to set up required time stamp. Following options are available:
	[1] No time shift
	[2] Time is shifting N days backward
	[3] Time is shifting N minutes backward
	where N - is integer number
	This function is very important. Real-time systems usually have a limitation on the time of incoming data. Some of them does not allow using "old" data, that is data with year or month or day not corresponding to the current one. Substitution of time stamp allows users to solve this problem.
Playback mode	Determines the method of job termination. If Type=1 (single pass playback) end of data means end of job, so when the last packet (out of all data files) is read passed to RTPD , the job is terminated. If Type=3 (multi pass playback), then after all data is read simref goes to the beginning of data. With this option, the time shift will be reset.

Table 1. Parameters shifted to simref

Note: There may be one or several source REF TEK data files. The sequence of their reading is determined by the sequence of their recording in the config file.

Configuration file structure

This is an example of the configuration file:

```
# Unit ID number (may be different from the real DAS ID).
Unit = 8021

#Packet delivery speed
# 0 - in-between packet interval corresponds to the real DAS work.
# N - the above interval is N seconds
Delay = 0

# Time shift:
# 0 - data time is not changed
#<0 - data time is shifted N days backward
#>0 - data time is shifted N minutes backward,
# N - is integer.
Start = -1

#Playback mode
# 0 - cycling, or multi pass playback
# 1 - single pass playback
# 2 - cycling mode with RTPD connection close/open after each pass
# 3 - single pass with request to continue.

Type = 0

# List of source REF TEK data files:
File = 8021.dat
File = 8022.dat
```

In current example one can see the simulation of work by the DAS with Unit ID = 8021 with real time data delivery with 1 day backward time shift. The REF TEK source data files are *8021.dat* and *8022.dat*. After these files are read (*8021.dat* first and *8022.dat* second) the program “rolls back” to the beginning of data files.

1.2.2 Data sample sequence simulation.

Process **simcon** allows the user to simulate the station work using source data files, containing the sequences of data samples, for instance CSS-3.0 waveform files with samples in format INT32.

Parameters description

The process attaches to **RTPD** and sends the data from the mentioned files. The following parameters may be used for the process configuration:

- Setting up Unit ID
- Setting up Stream number
- Setting up Channel number
- Setting up sampling rate
- PDS (real or fixed)
- Time shift

Process **simcon** creates REF TEK data packets and introduce data from the configuration file into these packets.

It is necessary to set up Unit ID, Stream # and Channel # for data packets to be compliant to REF TEK data packet format specification. It is also possible to use the same config file to simulate the whole network.

Parameter	Description
PDS	May be set to real, or with predefined time intervals. In the first case data packets will be delivered to RTPD according to the real delays (the delays are defined according to the time difference of neighboring packets in source files rounded to seconds). In the second case the in-between data packets interval is determined by the fixed time (in seconds).
Time shift	Allows the user to set up required time stamp. The following options are available:
	[1] No time shift
	[2] Time is shifting N days backward
	[3] Time is shifting N minutes backward
	where N - is an integer number
	Function is very important. Real-time systems usually have a limitation on the time of incoming data. Some of them does not allow using “old” data, that is data with year or month or day not corresponding to the current one. Substitution of time stamp allows users to solve this problem.

One data file must be assigned to each channel. It may be one file for all channels or different data files for different channels. After end of data program rolls back to the beginning and starts reading over, except for the cases $Type=\{1 | 3\}$, when the program either stops or request the keyboard input.

Configuration file description

Unit ID number that will be imitated:

Unit = 8021

#Packet delivery speed:

0 - in-between packet interval corresponds to the real DAS work.

N - the above interval is N seconds

Delay = 0

#Time shift:

0 - data time is not changed

#<0 - data time is shifted N days backward

#>0 - data time is shifted N minutes backward,

N - is integer.

Start = 5

#DAS Channel list:

Chan = Stream #, Channel #, Sampling_Rate, File Name

Chan = 1,0,40,css1.w

Chan = 1,1,20,css2.w

In this example we simulate a DAS with Unit ID 8021 and two channels (0 and 1) at Stream 1 in real-time packet delivery mode with time shift 5 minutes backward. The source files are *css1.w* and *css2.w* for first and second channels.

1.2.3 Simple data sequence generation

This regime is simulated by process **simsim**. Almost everything in this section is the same as in Table 1 on page 3. The only difference is that instead of reading data files, data samples are generated within the program as the integer number sequence, separately for each channel. Also, if **Type** = 3 then the program is terminated.

Section: 2 SIMARCH – REF TEK real-time data stream simulation program

The program is intended for reading data from REF TEK archive organized by **RTPD** and simulating real-time data streams to the designated **RTPD** (only **DT** packets are transferred). This program differs from the **SIMUL** program. **SIMUL** reads files fetched from the archive. **SIMARCH** does not require file extraction from the archive.

2.1 • Command format

The program is launched by the command:

simarch simarch.ini [start_time end_time]
where

Parameter	Description
simarch	name of the program
simarch.ini	name of the configuration file (may have another name)
start_time	If present with end_time must have YY:JJJ:HH:MM:SS format
end_time	If present with start_time must have YY:JJJ:HH:MM:SS format. May represent the requested time interval in seconds

Note: Time parameters may be omitted in a command line and represented in the *.ini* file. However, command line arguments override the ones in a configuration file.

The program may be terminated by Ctrl+C.

2.2 • Configuration file example:

```
#####
# SIMARCH config file
# Lines must be not more 255 bytes
# Lines with "#" or ";" in first position ignored (comments)
# RTPD = IP_address[:port_number]
#   Defined destination RTPD-server
#   IP_address may by nnn.nnn.nnn.nnn or hostname
#   If port_number not defined - set default (2543)
# Archive = path_to_archive
#   Defined path to RTPD archive (source data)
# Begin = YY:JJJ:hh:mm:ss
#   Defined begin of interval in archive
#   If Begin defined in command line - this parameter ignored
# End = YY:JJJ:hh:mm:ss
#   or
# End = nnn    (seconds)
#   Defined end of interval in archive
#   If End defined in command line - this parameter ignored
# Delay = 0 | -n
#   Defined real rate or acceleration
#   -1 equ 2 times, -2 equ 3 times, etc.
# Start = 0 | n_days | -n_minutes
#   Defined date/time in DT packets
#   If Start = 0 date/time in DT packets not changed
#   If Start = -n date in DT packets set to (current - n) days
#   If Start = n date/time in DT packets set to (current -n) minutes
# Type = 0 | 1
#   If Type = 0 - after end of job simarch asks what to do (answers: C to cancel, R
to
# restart, S to start the whole process over.
#   If Type = 1 - auto-restart
# Debug = 0 | 1 | 2 | 3
#   Defined debug messages level (0 - minimum)
# WaitInt = n
#   Defined wait interval (minutes) before restart
# Exclude = unit_id1, unit_id2, ...
#   Simulates all DAS(es) presented in archive but exclude list
# Include = unit_id1, unit_id2, ...
#   Simulates DAS(es) only from include list (if presented in archive)
#   If Exclude and Include not present - simulates all DAS(es)
#   If both Exclude and Include present - Exclude ignored
#   Several (not just one) lines may be used for Exclude or Include
#   (total is not more then 128 DAS)
#####
RTPD = localhost:2543
Archive = /Archive/imd3
Begin = 01:066:12:00:00
End = 01:068:12:00:00
Delay = -9
Start = 0
Type = 1
Debug = 1
WaitInt = 1
Stream = 1
#Exclude = 7953, 7966, 7955,7965
#Include = 7953
```

2.3 • Description of parameters:

Parameter	Description
RTPD	IP address of RTPD host, and a port number after delimiter (":", ",", or "/" allowed). If the port number is omitted, then the default port number is 2543
Archive	Complete path to archive
Begin	Data and time of the beginning of the requested interval (YY:JJ:hh:mm:ss)
End	Data and time, if the end of the requested interval (YY:JJ:hh:mm:ss), or the number of seconds of the requested interval.
Delay	Data transfer rate (simulated real-time or accelerated)
Start	Start time of the first packet (may be real, or taken from the packet, or changed).
Type	Data feeding mode (cycling, with restart, or single-pass mode with the request after each pass)
Type	A level of debugging messages (from 1 to 4) (optional)
WaitInt	A delay in minutes after each pass of playback.
Stream	DAS Stream number (default is 1).
Exclude	A list of DAS(s), not included to the simulation process (optional).
Include	A list of DAS(s), included into simulation process (optional).

1. If the **Stream** parameter field is present, data from this stream only will be used for simulation.
2. If the **Exclude** parameter is described like: **Exclude** = xxxx, yyyy, zzzz, then data from DAS(s) with numbers xxxx,yyyy, zzzz will **not** be used.
3. If the **Include** parameter is described like: **Include** = xxxx, yyyy, zzzz, then data from DAS(s) with numbers xxxx,yyyy, zzzz **ONLY** will be used.
4. If neither **Include**, nor **Exclude** are present, then ALL DAS(s) data is used. Several lines could be used to count the DAS(s) numbers. If both **Include** and **Exclude** are present, **Exclude** is ignored.
5. The DAS(s) list is defined by examining the archive directory.

Delay may have values of:

Delay of	Description
0	Real-time emulation
<0	Accelerated regime

Start gives the shift for packet times

Value	Description
0	Date and time remains original
>0	Beginning time is shifted backward on designated number of minutes (data, hour and minute is changed in simulated packets)
<0	Beginning time is shifted backward on designated number of days (data only is changed in the packets)

Note: In this design, seconds values remain the same.

Type gives the playback mode. If Type=0, then after all requested data playback (end of interval or end of data in archive) a user is prompted of the next action:

Repeat, Start or Cancel

Parameter	Description
C	The program is terminated.
R	The repeated playback of the same time interval is activated.
S	The new time interval is requested to be entered and its playback starts.

2.4 • Algorithm of work

The program builds the list of DAS(s), taking into account the **Include** and **Exclude** fields of the **.ini** file. It examines the archives subdirectories taking into account the **Begin** parameter. The file is searched containing the data packets for the requested **Stream** and **Begin**. If the relevant file is not found (in a subdirectory, containing the date of the beginning in its name), then the search is performed in a subdirectory with the previous date. Then the packet with the relevant beginning time is searched inside the file found. The data beginning time in this packet is then inserted into a DAS table. Then the cycle of the packet sending starts. After the packet is sent, the next packet is read and its time is defined. This cycle continues until the end of the file or end of the time interval (**End** field). If the file is over but the interval is not, then the next file is searched in same subdirectory. If such a file is not found there, then next date is processed. If the file, related to the **Begin** is not found, then the nearest file is processed.

Section: 3 Installation and Operation

3.1 • Introduction

1. Create a directory named **SIMUL** on your C drive.
2. Copy files from the CDROM to **SIMUL** directory.
3. Choose the type of simulation program you will use:

Program	Purpose
SIMUL	Works with files fetched from the REF TEK archive
SIMARCH	Works with the entire archive

3.2 • Configure the application

1. Edit your configuration files according to the **SIMUL** or **SIMARCH** description.
2. Prepare archive or data files for processing.
3. Edit your configuration files to supply the correct paths to the data
4. Adjust port numbers in the *RTPD.ini* file according to the ones given in simulation programs and in your client data processing application (for example, **Earthworm**).

For example, if your **SIMUL** configuration file *simctrl.ini* contains the following:

```
RTPD = localhost, 8031
```

then the *RTPD.ini* file must have the same UDP port specified:

```
UDPport      8031
```

3.3 • Start the application

1. Start the **RTPD** service or application that applies to the proper *RTPD.ini* file.
2. Start the simulation program, selected above (**SIMUL** or **SIMARCH**) according to the program description.

For example, to start **SIMUL**, you can type in your MS-DOS window:

```
simctrl
```

The program will read *simctrl.ini* (default) and start.

Note: **simctrl** can also be started with the configuration file: **simctrl config.ini**

Remember, that the partial DAS configuration files, utilized by the **simref** modules (child **simctrl** processes) must be described in *simctrl.ini* file like this:

```
DAS = ./simref, 8020.ini
```

3.4 • Diagnostics to check data path

1. The program will give you an output (diagnostics) like this as it initializes.

```
RTPD=localhost,8031 NDAS=1
Started ./simref(8020.ini)
----->8020(./simref) started Wed Oct 10 11:57:00 2001
8020 seek RTPD: 127.0.0.1/8031
8020 RTPD: 192.168.100.75/8031
8020 RTP Link Synchronization
8020 Link Synchronization - OK
8020 open file data/1.dat
8020 EH ORIG:Sat Jul 15 06:03:02 2000
8020(0) Wed Oct 10 11:57:02 2001
8020(1) Wed Oct 10 11:57:02 2001
8020(2) Wed Oct 10 11:57:02 2001
8020(0) Wed Oct 10 11:57:11 2001
```

This can be interpreted as:

- The **RTPD** UDP port number is 8031, and it's functioning on a host with the IP address of 192.168.100.75,
 - Just one DAS is in the simulation process, and the Unit ID is 8020 and it has 3 channels supplying data (0,1,2),
 - The data files is 1.dat in the directory data next to current.
2. The messages will be output to the screen to supply a status how the process is going.
 3. To test how data is passed to the application, start the **rtpc** application. This a simple program showing **REF TEK** data packets coming from the **RTPD** client port:

rtpc port=8030

The program give this output to the console:

```
Requested attributes:
DAS 'mask'          (at_dasid) = 0
Packet mask         (at_pmask) = 0x0070
Stream mask         (at_smask) = 0x00ff
Socket I/O timeout  (at_timeo) = 30
TCP/IP transmit buffer (at_sndbuf) = 0
TCP/IP receive buffer (at_rcvbuf) = 0
blocking I/O flag   (at_block) = TRUE

Initiating connection to server at localhost:8030
connected to localhost:8030
Actual parameters:
DAS 'mask'          (at_dasid) = 0
Packet mask         (at_pmask) = 0x0070
Stream mask         (at_smask) = 0x00ff
Socket I/O timeout  (at_timeo) = 30
TCP/IP transmit buffer (at_sndbuf) = 0
TCP/IP receive buffer (at_rcvbuf) = 0
blocking I/O flag   (at_block) = TRUE
DT 001 08020 00001 2001:283-16:57:02.536 00007 00 00 00890 C0
DT 001 08020 00002 2001:283-16:57:02.536 00007 00 01 00890 C0
DT 001 08020 00003 2001:283-16:57:02.536 00007 00 02 00890 C0
DT 001 08020 00004 2001:283-16:57:11.436 00007 00 00 00892 C0
DT 001 08020 00005 2001:283-16:57:11.436 00007 00 01 00892 C0
DT 001 08020 00006 2001:283-16:57:11.436 00007 00 02 00892 C0
DT 001 08020 00007 2001:283-16:57:20.356 00007 00 00 00892 C0
```

3.5 • Getting started

With the data coming to **RTPD** (from watching **SIMUL** diagnostics) and from **RTPD** (from watching **rtpc** diagnostics) verified:

1. Start **Earthworm** with the data port assigned according to the client port described in *RTPD.ini* file:
 Port 8030

2. Start **SIMARCH** with the configuration file name as an argument.

Note: If no argument is given a short help message will appear, as shown below:

>**simarch**

```
Usage: simarch file_cfg [begin end]
Format begin: YY:JJJ:hh:mm:ss end: YY:JJJ:hh:mm:ss or end:nnn (sec)
If begin-end not defined - used from file_cfg
```

Started with the configuration file argument, **SIMARCH** gives messages like this:

simarch imd.ini

```
7965 open previous date dir
```

```
*****
*SIMARCH started. ARCHIVE:/export/home/mike/IMD/imd DELAY=-100 STARTTIME=-1
*      BEGIN:Wed Mar  7 23:43:46 2001
*      END  :Thu Mar  8 00:03:46 2001
*****
```

```
7953 open file /export/home/mike/IMD/imd/2001066/7953/1/232400001_0036EE94
7954 open file /export/home/mike/IMD/imd/2001066/7954/1/233349991_0036F09C
7955 open file /export/home/mike/IMD/imd/2001066/7955/1/234346153_0036EE80
7957 open file /export/home/mike/IMD/imd/2001066/7957/1/232352923_0036EE80
7960 open file /export/home/mike/IMD/imd/2001066/7960/1/233350254_0036F09C
7961 open file /export/home/mike/IMD/imd/2001066/7961/1/232917834_0036EE80
7965 open file /export/home/mike/IMD/imd/2001066/7965/1/234434777_0036EE80
7966 open file /export/home/mike/IMD/imd/2001066/7966/1/231049191_0036EEA8
7961 ORIG: Wed Mar  7 23:43:33 2001
7961 seek RTPD: 127.0.0.1/8031
7961 RTPD: 192.168.100.75/8031
7961 RTPD Link Synchronization
7961 Link Synchronization - OK
7961 SEND: Tue Oct  9 23:43:33 2001
7954 ORIG: Wed Mar  7 23:43:36 2001
7954 seek RTPD: 192.168.100.75/8031
7954 RTPD: 192.168.100.75/8031
7954 RTPD Link Synchronization
7954 Link Synchronization - OK
7954 SEND: Tue Oct  9 23:43:36 2001
7960 ORIG: Wed Mar  7 23:43:38 2001
7960 seek RTPD: 192.168.100.75/8031
```


3. After all DAS streams are synchronized, there will be a sequence of messages like the following:

```
7955 SEND: Tue Oct 9 23:44:05 2001
7966 ORIG: Wed Mar 7 23:44:06 2001
7966 SEND: Tue Oct 9 23:44:06 2001
7953 ORIG: Wed Mar 7 23:44:07 2001
7953 SEND: Tue Oct 9 23:44:07 2001
7961 ORIG: Wed Mar 7 23:44:09 2001
7961 SEND: Tue Oct 9 23:44:09 2001
7954 ORIG: Wed Mar 7 23:44:11 2001
7954 SEND: Tue Oct 9 23:44:11 2001
7954 ORIG: Wed Mar 7 23:44:12 2001
7954 SEND: Tue Oct 9 23:44:12 2001
7957 ORIG: Wed Mar 7 23:44:12 2001
7957 SEND: Tue Oct 9 23:44:12 2001
7960 ORIG: Wed Mar 7 23:44:14 2001
7960 SEND: Tue Oct 9 23:44:14 2001
7960 ORIG: Wed Mar 7 23:44:14 2001
```

3.5.1 Description of messages

SIMARCH started. ARCHIVE:/export/home/mike/IMD/imd DELAY=-100

The source archive is situated in the directory */export/home/mike/IMD*, the archive name is *imd*, the acceleration of data delivery to RTPD is 100 (100 times as faster than the acquisition rate).

```
7953 open file /export/home/mike/IMD/imd/2001066/7953/1/232400001_0036EE94
```

Data acquired by DAS with Unit ID 7953 found in archive, reading file 232400001_0036EE94

```
7961 ORIG: Wed Mar 7 23:43:33 2001
7961 seek RTPD: 127.0.0.1/8031
7961 RTPD: 192.168.100.75/8031
7961 RTPD Link Synchronization
7961 Link Synchronization - OK
7961 SEND: Tue Oct 9 23:43:33 2001
```

Establishing synchronization with RTPD for the DAS with Unit ID 7961

```
7966 ORIG: Wed Mar 7 23:44:06 2001
7966 SEND: Tue Oct 9 23:44:06 2001
```

Original packet is dated Wed Mar 7 23:44:06 2001

Processed packet differs from the original one with the time, set to Tue Oct 9 23:44:06 2001.

3.5.2 Example of the configuration file:

Corresponding to this protocol:

```
LogFile=simarch.log
RTPD = localhost:8031
Archive = /export/home/mike/IMD/imd
Begin = 01:066:23:43:46
End = 01:067:00:03:46
Delay = -100
Start = -1
Type = 0
Debug = 1
WaitInt = 1
```