NerveNet DB synchronization

Record Transmission Instructions

1st Edition

2021 Year January 9

Nashua Solutions Co., Ltd.

Reference number: DP\_NV\_SP0128-10

Revision history

| edition | Date Modifier | Details of the revision |
| --- | --- | --- |
| 1.0 | 2021. 11. 9  Takumi Nakamura | New |

**Contents**

[1. Introduction 4](#_Toc112139007)

[2. Related diagrams 4](#_Toc112139008)

[3. Transmission example 5](#_Toc112139009)

[3.1. Send and receive with the hearsayif tool 5](#_Toc112139010)

[3.1.1 Standby at receiver 5](#_Toc112139011)

[3.1.2 Send record on sender 6](#_Toc112139012)

[3.1.3 Show on receiver 7](#_Toc112139013)

[3.2. Send with the hearsayif tool 8](#_Toc112139014)

[3.3. Check Directly on Rx BS 8](#_Toc112139015)

[3.4. Receive Data Through hearsayif-monitor tool 9](#_Toc112139016)

[Standby at receiver 9](#_Toc112139017)

[3.5. Write record directly to the Sender BS 9](#_Toc112139018)

[3.5.1 Show record on Receiver BS 10](#_Toc112139019)

[4. Related diagram description 11](#_Toc112139020)

[4.1. database 11](#_Toc112139021)

[4.2. hearsayif tool 11](#_Toc112139022)

[4.3. Hearsay I/F 11](#_Toc112139023)

[4.4. DB synchronization 11](#_Toc112139024)

# Introduction

This document describes NerveNet's base station-to-base database synchronization mechanism (DB synchronization). Describes the process of record transmission.

| phrase | explanation |
| --- | --- |
| DB synchronization between base stations | Between NerveNet base stations and NerveNet terminals  A mechanism that enables per-record DB synchronization.  It also provides a file synchronization mechanism that hangs on records. |

# Related diagrams

Shows a related diagram of record transmission in DB synchronization.



In the figure, Hearsay Daemon and hearsayif-publish, hearsayif-monitor tool are the programs inside the base station container.

The body of the record is stored in the **database** in the diagram.

# Transmission example

Examples of transmission by tools and user programs are shown. Use box information sharing table **shbt\_boxshare** for BOX synchronization as the table to be transmitted.

## Send and receive with the hearsayif tool

Send a record with the hearsayif-publish tool in the base station container.

It is also configured to wait for records to be received by the hearsayif-monitor tool.



### Standby at receiver

First, at the receiving base station BS31, use the hearsayif-monitor tool to listen.

The following commands to launch hearsayif-monitor in the base station container are:

* docker exec -it tsg /writable/sbin/hearsayif-monitor shbt\_boxshare

The container name is tsg and the table name is shbt\_boxshare.

The following contents are displayed and will be in a standby state.

----

[main]

start main loop

[ThreadHearsayIf::udStream\_Setup]

udpath=/var/tmp/ud-hearsayd

[UnixProto Session::Connect]

udn=to-hearsay-if sess-conn stmphase=1

[UnixProto ForHearsayIf::Client::OnConnect]

udn=to-hearsay-if

[ForHearsayIf::Client::Cmd\_Subscribe]

send subscribe table=shbt\_boxshare

[ForHearsayIf::Client::answerCommon]

udn=to-hearsay-if id\_action=1 answer=subscribe status=COMPLETE

----

### Send record on sender

Prepare the record to send as file share.txt in the receiving base station BS30.

The file contents are as follows:

<record>

<table>shbt\_boxshare</table>

<id\_box>414243</id\_box>

<body>granbelm</body>

</record>

Send with the hearsayif-publish tool.

The following commands to launch hearsayif-publish in the base station container are

* cat share.txt | docker exec -i tsg /writable/sbin/hearsayif-publish

The container name is tsg. The argument does not require a table name.

*The option for docker exec is -t, not -it.*

*If you specify -i, the redirect does not work, so-i should be removed.*

The following contents will be displayed and the transmission instruction will be completed.

----

[main]

start main loop

[ThreadHearsayIf::udStream\_Setup]

udpath=/var/tmp/ud-hearsayd

[UnixProto Session::Connect]

udn=to-hearsay-if sess-conn stmphase=1

[UnixProto ForHearsayIf::Client::OnConnect]

udn=to-hearsay-if

[ForHearsayIf::Client::Cmd\_Write]

send write table=shbt\_boxshare

[ForHearsayIf::Client::answerWrite]

id-action=2 table=shbt\_boxshare id\_record=017d0256bb730222001e status=COMPLETE reason=

----

### Show on receiver

On the receiving base station BS31, the display of the hearsayif-monitor tool is updated as follows:

Record: After the content of the received record.

----

[main]

start main loop

[ThreadHearsayIf::udStream\_Setup]

udpath=/var/tmp/ud-hearsayd

[UnixProto Session::Connect]

udn=to-hearsay-if sess-conn stmphase=1

[UnixProto ForHearsayIf::Client::OnConnect]

udn=to-hearsay-if

[ForHearsayIf::Client::Cmd\_Subscribe]

send subscribe table=shbt\_boxshare

[ForHearsayIf::Client::answerCommon]

udn=to-hearsay-if id\_action=1 answer=subscribe status=COMPLETE

Record:

<body>granbelm</body>

<id\_box>414243</id\_box>

<id\_node\_update>0222001e</id\_node\_update>

<id\_record>017d0256bb730222001e</id\_record>

<time\_discard>1636508178291</time\_discard>

<time\_update>1636421778291</time\_update>

<timestamp\_sync>2021-11-09 01:36:35</timestamp\_sync>

----

* id\_record
* time\_udpate
* time\_discard

do not exist at the time of transmission and are increasing.

It is not hearsayif-publish that is generating them.

In Hearsay Daemon, it complements the acceptance of transmission instructions in Hearsay I/F.

## Send with the hearsayif tool

Send records with the hearsayif-publish tool located in the base station container (BS30).

This diagram shows a configuration that checks that the message has been received by checking the database directly on the receiving BS (BS31).



The receiver does not need any waiting.

The sending procedure is the same as send and receive [records on the sender side](#_送信側でレコード送信) of the [hearsayif](#_hearsayifツールで送受信) tool. (as mentioned above).

## Check Directly on Rx BS

On the receiving base station BS31, use the sqlite3 command to browse the database directly.

First, use the command below to open the database.

**sqlite3 /var/tmp/hearsayd.sqlite3**

Run the SQL command from the sqlite3 tool prompt to browse to the records in the shbt\_boxshare table.

**select body, hex(id\_box), hex(id\_node\_update), hex(id\_record), time\_discard, time\_update, timestamp\_sync from shbt\_boxshare;**

The contents of the record are displayed as follows:

**granbelm|414243|0222001E|017D0256BB730222001E|1636508178291|1636421778291|2021-11-09 01:36:35**

## Receive Data Through hearsayif-monitor tool

The sender writes records directly to the database.

This configuration waits for record reception with the hearsayif-monitor tool in the base station container.



### Standby at receiver

On the receiving base station BS31, use the hearsayif-monitor tool to listen.

* docker exec -it tsg /writable/sbin/hearsayif-monitor shbt\_boxshare

## Write record directly to the Sender BS

Write the record that you want to send directly to the database. For information about the composition and prerequisites for records, see the Record Operations Manual. See docx.

Connect to the database in sqlite3 /var/tmp/hearsayd.sqlite3

For example, write a record with the following SQL command:

Insert into shbt\_boxshare (body, id\_box, id\_node\_update, id\_record, time\_discard, time\_update) VALUES ('white lily', 'ABC', x'021e', x'0123456789abcdef', 1636434035000, 1636430435000);

insert into shbt\_boxshare (attached, body, id\_box, id\_node\_update, id\_record, time\_discard, time\_update ) VALUES ( '/path/to/somewhere.mov', 'white lily', 'ABC', x'021e', x'0123456789abcdef', 1636434035000, 636430435000);

If time\_update and time\_discard are not specified at the appropriate time, D-B synchronization will not work.

### Show record on Receiver BS

On the receiving base station BS31, the display of the hearsayif-monitor tool is updated as follows:

Record: After the content of the received record.

----

[main]

start main loop

[ThreadHearsayIf::udStream\_Setup]

udpath=/var/tmp/ud-hearsayd

[UnixProto Session::Connect]

udn=to-hearsay-if sess-conn stmphase=1

[UnixProto ForHearsayIf::Client::OnConnect]

udn=to-hearsay-if

[ForHearsayIf::Client::Cmd\_Subscribe]

send subscribe table=shbt\_boxshare

[ForHearsayIf::Client::answerCommon]

udn=to-hearsay-if id\_action=1 answer=subscribe status=COMPLETE

Record:

<body>white lily</body>

<id\_box>414243</id\_box>

<id\_node\_update>021e</id\_node\_update>

<id\_record>0123456789abcdef</id\_record>

<time\_discard>1636434035000</time\_discard>

<time\_update>1636430435000</time\_update>

<timestamp\_sync>2021-11-09 04:03:05</timestamp\_sync>

----

This time, the sender does not use Hearsay I/F of Hearsay Daemon.

All required fields, such as id\_record, are specified when writing to the database.

# Related diagram description

This is the description of the related diagram at the beginning.



## database

The Database in the figure is a SQLite3 file in the nervenet-tsg-sqlite container.

the nervenet-tsg-mysql container is a MariaDB database.

## hearsayif tool

hearsayif-publish, hearsayif-monitor tool description

HearsayIf tool. See docx.

It is a test tool built-in base station container.

## Hearsay I/F

The description of Heaersay I/F connecting Hearsay Daemon and heasayif tools

See Hearsay daemon IF.docx.

In-bureau-only interface on Unix domain.

## DB synchronization

The communication between Hearsay Daemon in the figure is DB synchronization.

Records from adjacent stations are captured from each other on the base station-to-station link.

As a result, all base stations retain the same records.

from abhimanyu pandey to everyone: 2:13 PM

insert into shbt\_boxshare ( body, id\_box, id\_node\_update, id\_record, time\_discard,

time\_update) VALUES ( 'white lily', 'ABC', x'021e', x'0123456789abcdef', 1636434035000,

1636430435000);

from abhimanyu pandey to everyone: 2:13 PM

i need an example for a real file

from abhimanyu pandey to everyone: 2:13 PM

e.g. jpg file, etc.

from 泰伯 大和田 to everyone: 2:15 PM

insert into shbt\_boxshare ( attached, body, id\_box, id\_node\_update, id\_record, time\_discard,

time\_update ) VALUES ( '/path/to/somewhere.mov', 'white lily', 'ABC', x'021e', x'0123456789abcdef', 1636434035000, 1636430435000);

from abhimanyu pandey to everyone: 2:22 PM

schema name, username, password

from abhimanyu pandey to everyone: 2:22 PM

just to understand the architecture

from abhimanyu pandey to everyone: 2:22 PM

which columns exist, etc.

from abhimanyu pandey to everyone: 2:22 PM

and then when we update the db we can do the confirmation with the admin

from 中村 匠 to everyone: 2:42 PM

I confirm my environment.'docker exec -i' is correct. not use '-t'.

from 泰伯 大和田 to everyone: 2:43 PM

Thank you, Nakamura san.