eProsima Non-Intrusive DDS Recorder

User Manual
Version 1.0.0

The Middleware Experts

eProsima
Proyectos y Sistemas de Mantenimiento SL
Ronda del poniente 16 – Bajo K
28760 Tres Cantos Madrid
Tel: + 34 91 804 34 48
info@eProsima.com – www.eProsima.com

Trademarks

eProsima is a trademark of *Proyectos y Sistemas de Mantenimiento SL*. All other trademarks used in this document are the property of their respective owners.

License

eProsima Non-Intrusive DDS Recorder is licensed under the terms described in the DDSRECORDER_LICENSE file included in this distribution.

Technical Support

- Phone: +34 91 804 34 48
- Email: support@eprosima.com

Contents

1 Introduction

eProsima Non-Intrusive DDS Recorder is a tool to record all the DDS traffic in your network, using a non-intrusive mechanism allowing you to test, analyze or log your DDS distributed system without adding any new DDS participant or service, ensuring you are recording the real behavior and timing.

1.1 How it works

eProsima Non-Intrusive DDS Recorder records the DDS traffic sniffing the DDS protocol (RTPS) through the switch debug port. The tool dissects the protocol and builds a complete database of all the DDS entities (Participants, Publishers, Subscribers, and Topics), the Data Types, and all the exchanged messages.

eProsima Non-Intrusive DDS Recorder does not record just raw data: it builds a message table for each DDS Topic, with the same fields as the corresponding DDS Topic Data Type.

In this release you need a sniffer such as wireshark to save the network traffic into a standard packet capture file (PCAP). *eProsima Non-Intrusive DDS Recorder* will later parse the file to translate the RTPS messages into a human readable format and store them in a SQLite database. In order to get all the traffic in your network you should sniff your switch debug port, otherwise you would get just the traffic directed to the node where the sniffer is running. Future releases will include a built-in sniffer to process the network packets in real time.

DDS uses an automatic discovery process to discover all the DDS entities in your network including your topic data types. *eProsima Non-Intrusive DDS Recorder* analyzes the discovery traffic to build a set of tables in a database using your data types schema to store the DDS user data traffic later.

To get the data type information *eProsima Non-Intrusive DDS Recorder* searchs for the data type definition (*Typecode*) in the discovery messages. It is important to note that not all the available DDS implementations send the *typecode* information (see supported DDS implementations in the release notes). Future releases of *eProsima Non-Intrusive DDS Recorder* will allow the use of an IDL file to generate the *typecode* of your types.

2 Usage

eProsima Non-Intrusive DDS Recorder is a command line application. The command line syntax is:

```
DDSRecorder <pcapFile> [-db <database>] [-tcMaxSize <size>] [-idl <file>] [-help]
```

- *<pcapFile>*: name of the file that the application will analyze. This file should be a PCAP format.
- *-db <database>:* name of the SQLite file that will be created and used to store the translated RTPS messages. By default *eProsima Non-Intrusive DDS Recorder* creates the file *dump.db*.
- -tcMaxSize <size>: TypeCode maximum allowed size (Default: 2048)
- -idl <file>: IDL file containing all data types used in the captured system if their typecodes are not being sent in the discovery phase.
- -help: Print help information.

3 Generated Database Structure

eProsima Non-Intrusive DDS Recorder creates a set of tables for the discovery information and the user data traffic

- **Discovery Tables**: Used to store relevant discovery information.
  - _topics Table: Stores the DDS Topics and their data type information
  - _endpoints Table: Stores DDS Endpoint information (Datareaders and Datawriters)
  - _endpointsDiscoveryMessages Table: Stores DDS Endpoint discovery messages.
- **User Topics Tables:**
  - **Topic Main Tables**: A table per DDS topic storing the topic messages using its data type schema.
    - topicName1
    - topicName2
    - …
    - topicNameN
  - **Topic Auxiliary Tables**: If a DDS topic contains a variable length field, such as an array or sequence, an auxiliary table is created to store the field values of each topic message.
    - TopicNameT
      - TopicNameT_varLengthFieldName1
      - TopicNameT_varLengthFieldName2
      - …
      - TopicNameT_varLengthFieldNameN

3.1 Discovery Tables

3.1.1 _topics table

eProsima Non-Intrusive DDS Recorder creates a table named "_topics". This table stores information about all DDS Topics found in the sniffer trace.

Table 1: _topics Table Fields

Table field	Field type	Description
topic_name	VARCHAR(255)	The name of the discovered DDS Topic
type_name	VARCHAR(255)	The name of the DDS Topic data type
typecode	TEXT	The *Typecode* of the data type in a human readable format.

3.1.2 _endpoints table

eProsima Non-Intrusive DDS Recorder creates a table named "_endpoints". This table stores information about the *Datawriters* and *Datareaders* found in the sniffer trace.

Table 2: _endpoints Table Fields

Table field	Field type	Description
rtps_host_id, rtps_app_id, rtps_instance_id, rtps_entity_id	UNSIGNED INT	These four fields contain the unique identifier for each discovered entity.
endpoint_type	CHARACTER(10)	Type of entity: *Datareader* or *Datawriter*.
topic_name	VARCHAR(255)	Entity associated *topic*

3.1.3 _endpointDiscoveryMessages table

eProsima Non-Intrusive DDS Recorder creates a table named "_endpointDiscoveryMessages". This table stores all RTPS messages involved in the endpoint discovery phase.

Table 3: _endpointDiscoveryMessages Table Fields

Table field	Field type	Description
message_id	INT	Numeric Identifier for the message matching the sniffer packet number.
sniffer_timestamp_sec sniffer_timestamp_usec	INT	Sniffer Timestamp (Seconds, Microseconds).
ip_src	VARCHAR(15)	Source IP address.
ip_dst	VARCHAR(15)	Destination IP address.
src_rtps_host_id, src_rtps_app_id, src_rtps_instance_id	UNSIGNED INT	RTPS GUID (Global Unique ID) of the source participant.
src_timestamp_sec src_timestamp_nanosec	INT	Source time stamp of the discovery message (Seconds, Nanoseconds): This timestamp is set when the source participant sends the discovery message using its own clock.
dst_rtps_host_id, dst_rtps_app_id, dst_rtps_instance_id	UNSIGNED INT	RTPS GUID (Global Unique ID) of the destination participant. These fields could be empty if the discovery message is not delivered to only one *DomainParticipant*.
endpoint_rtps_entity_id	UNSIGNED INT	rtps_entity_id of the endpoint. The Endpoint GUID is obtained appending this ID to the Source Participant GUID
endpoint_type	CHARACTER(10)	This field specifies if the endpoint is a *Datawriter* or a *Datareader*.
topic_name	VARCHAR(255)	The name of the *DDS Topic* associated with the *Datawriter* or *Datareader*
contains_typecode	UNSIGNED TINYINT	'1' if the information of the endpoint contains the *Typecode* of the Topic Data Type, otherwise '0'.

3.2 User Topics tables

3.2.1 Topic Main Tables

For each discovered DDS Topic*, eProsima Non-Intrusive DDS Recorder* creates a table named as the topic. The next invalid characters in the table's name are replaced by the character '_':

```
':' , '.' , '-'
```

This table stores all data samples of the *Topic*, using the following schema:

- Protocol Metadata fields
- Topic Data Type fields

Table 4: Topic Table Protocol Metadata fields

Table field	Field type	Description
message_id	INT	Numeric Identifier for the message matching the sniffer packet number.
sniffer_timestamp_sec, sniffer_timestamp_usec	INT	Sniffer Timestamp (Seconds, Microseconds).
ip_src	VARCHAR(15)	Source IP address.
ip_dst	VARCHAR(15)	Destination IP address.
src_rtps_host_id, src_rtps_app_id, src_rtps_instance_id	UNSIGNED INT UNSIGNED INT UNSIGNED INT	RTPS GUID (Global Unique ID) of the source participant.
src_timestamp_sec, src_timestamp_nanosec	INT INT	Source time stamp of the message (Seconds, Nanoseconds): This timestamp is set when the source participant sends the message using its own clock.
dst_rtps_host_id dst_rtps_app_id dst_rtps_instance_id	UNSIGNED INT UNSIGNED INT UNSIGNED INT	RTPS GUID (Global Unique ID) of the destination participant. These fields could be empty if the discovery message is not delivered to only one *DomainParticipant*.

3.2.2 Topic Auxiliary Tables

If a topic contains a variable length field, such as an array or sequence, an auxiliary table is created to store the field values of each topic message. You can get the complete information of a sample using a SQL Query to combine these auxiliary tables with the corresponding main tables. Check in the supported types chapter the Sequences & Arrays section, for more information

4 Supported data types

This section describes the data types that *eProsima Non-Intrusive DDS Recorder* supports and how the data is stored.

4.1 Basic types

For each basic type field in a DDS Topic*, eProsima Non-Intrusive DDS Recorder* creates a field in the corresponding Topic table using the same name and a compatible SQLite type.

Table 5: IDL to SQLite Type Mapping (Basic Types)

IDL Basic type	**SQL Field type**
octet	TINYINT
short	SMALLINT
unsigned short	SMALLINT UNSIGNED
long	INT
unsigned long	INT UNSIGNED
long long	BIGINT
unsigned long long	BIGINT UNSIGNED
char	CHARACTER(1)
string	TEXT
float	FLOAT
double	DOUBLE
boolean	TINYINT
enumeration	TEXT

4.2 Sequences & Arrays

eProsima Non-Intrusive DDS Recorder supports sequences and multidimensional arrays of basic types. Future releases will include support for sequences and arrays of user types.

For each sequence/array *eProsima Non-Intrusive DDS Recorder* will create:

- An **integer field** in the Topic Table named **<Array/SequenceName>_id**
- An **auxiliary table** named **<TopicName>_<Array/SequenceName>** to store the Array/Sequence Data. The table schema will be:
  - The **<Array/SequenceName>_id** integer field to identify the array/sequence.
  - A set of integer fields named <index_n> for the array/sequence indexes.
  - The fields for the data.

To view the topic samples data including an Array/Sequence field you can use an SQL Query:

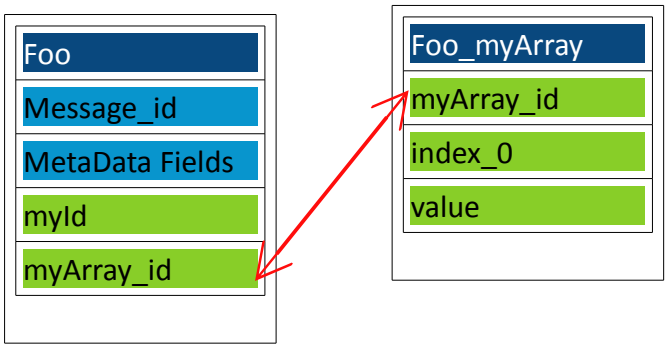
```
select * from <TopicName> inner join <TopicName>_<Array/SequenceName> on
      <TopicName>.<Array/SequenceName>_id =
      <TopicName>_<Array/SequenceName>.<Array/SequenceName>_id;
```

4.2.1 Example:

Consider a Foo Topic with the following Data Type

```
struct Foo_Type {
       long myId;
       long myArray[10];
};
```

eProsima Non-Intrusive Recorder would generate the following Tables:



To get the values of myArray for a given Foo sample you could use the Query:

```
select Foo_myArray.* from Foo inner join Foo_myArray on
       Foo.myArray_id = Foo_myArray.myArray_id
where
       Foo.message_id = <a_Valid_Id>;
```

4.3 Inner Structures

In the case of inner Structures, *eProsima Non-Intrusive DDS Recorder* creates a new field in the Topic table for each field in the inner structure prefixing the field name with the inner structure name:

```
<InnerStructure Name>_<Field Name>
```

4.3.1 Example:

Consider the following IDL

```
struct MyInnerStruct {
       long counter;
       string message;
};

struct MyStruct {
       long id;
       MyInnerStruct myIS;
}; /* This is the Type we will use for the topic */
```

eProsima Non-Intrusive DDS Recorder will create the following fields for the topic table:

Table 6: Inner Structure Example - Topic Fields

Table field	Field type
Message_id	INT
MetaData Fields	...
id	INT
myIS_counter	INT
myIS_message	TEXT

4.4 Unions

For Unions *eProsima Non-Intrusive DDS Recorder* creates a new field in the Topic table for each field in the union prefixing the field name with the union name:

<Union Name>_<Field Name>

eProsima Non-Intrusive DDS Recorder creates also a discriminator field for the union:

<Union Name>_discriminator

The discriminator specifies the union field used for a sample.

4.4.1 Example:

Consider the following IDL:

```
union MyUnion switch (long /* discriminator type */) {
    case 1:
        long counter;
    case 2:
        string message;
};

struct MyStruct {
       long id;
       MyUnion myU;
};/* This is the Type we will use for the topic */
```

eProsima Non-Intrusive DDS Recorder will create the following fields for the topic table:

Table 7: Union Example - Topic Fields

Table field	Field type
Message_id	INT
MetaData Fields	...
id	INT
myU_discriminator	INT
myU_counter	INT
myU_message	TEXT

5 HelloWorld example

This section will explain the usage of *eProsima Non-Intrusive DDS Recoder* through a simple example located at

```
[Recorder Install dir]/examples/HelloWorld
```

The folder contains the IDL file used for the data type, a sample PCAP file containing the sniffed traffic from a simple DDS Publisher and Subscriber applications, and the generated SQLite database. We will use the following IDL:

```
struct HelloWorld {
   long counter;
   string message;
};
```

To browse the SQLite database many Graphical Interfaces are available. The screenshots used for this section are taken using the SQLiteman GUI Tool. It can be downloaded freely from:

http://sqliteman.com/

5.1 Generating the applications and sniffing some network packets

Most DDS implementations count with an IDL compiler to generate Type Support for our data type and an example pub-sub applications. Run the applications, save some network traffic using a network sniffer, and save the results using the PCAP format. There are many network sniffer available, being WireShark the most known (http://www.wireshark.org/)

If you want to skip this step you can use the included sample PCAP file.

5.2 Generating the SQLite database

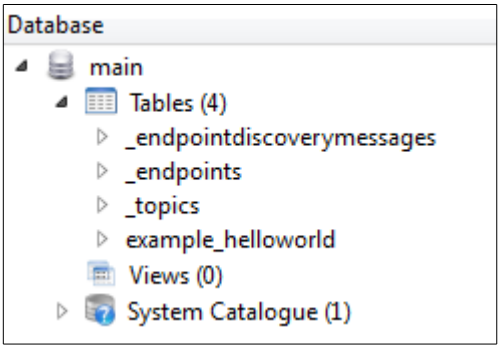
eProsima Non-Intrusive DDS Recorder will be used to generate the SQLite database to store captured RTPS messages in a human readable format. To create the database run this command:

```
DDSRecorder -db HelloWorld.db HelloWorld.pcap
```

Remember to have the appropriate permissions to create files in that folder.

5.3 Understanding the SQLite database

eProsima Non-Intrusive DDS Recorder creates four tables in the generated SQLite database:



These tables follow the schema described in the Generated Database Structure section.

5.3.1 _EndpointDiscoveryMessages table

eProsima Non-Intrusive DDS Recorder found two Endpoint discovery RTPS messages from the Publisher and Subscriber applications. The _EndpointDiscoveryMessages table shows these entries corresponding to a DDS DataWriter and a DDS DataReader:

	id	dst_rtps_instance_id	endpoint_rtps_entity_id	endpoint_type	topic_name	contains_typecode
1		{null}	2147483651	DataWriter	Example HelloWorld	1
2	)4	1	2147483652	DataReader	Example HelloWorld	1

From the discovery messages *eProsima Non-Intrusive DDS Recorder* extracts the information about endpoints and topics, creating the next two tables.

5.3.2 _Endpoints table

The _endpoints table shows the two DDS endpoints detected in the discovery traffic, a DataReader and a Datawriter.

	rtps_host_id	rtps_app_id	rtps_instance_id	rtps_entity_id	endpoint_type	topic_name
1	3232235788	6404	1	2147483651	DataWriter	Example HelloWorld
2	3232252933	5528	1	2147483652	DataReader	Example HelloWorld

5.3.3 _Topics table

The _Topics Table shows just one topic in this case with a human-readable representation of the type code, an IDL structure.

	topic_name	type_name	typecode

1	Example HelloWorld	HelloWorld	struct HelloWorld {
			long counter;
			string message;
			};

5.3.4 Example_HelloWorld table

In this example our topic name is "Example_HelloWorld" and *eProsima Non-Intrusive DDS Recorder* generates a table for the topic containing the samples data and some metadata:

	stamp_nanosec	dst_rtps_host_id	dst_rtps_app_id	dst_rtps_instance_id	counter	message
8	725840128	{null}	{null}	{null}	6	HelloWorld 6
9	730135040	{null}	{null}	{null}	7	HelloWorld 7
10	730135040	{null}	{null}	{null}	8	HelloWorld 8
11	738724864	{null}	{null}	{null}	9	HelloWorld 9
12	738724864	{null}	{null}	{null}	10	Hello
13	738724864	{null}	{null}	{null}	11	HelloWorld 11

HelloWorld