RTI DDS Spy

Version 7.1.0

Contents

Chapter 1

Introduction

RTI® DDS Spy (*rtiddsspy*) is invaluable when debugging thorny problems with DDS applications. It is a command-line tool that receives all DDS communication. *rtiddsspy*:

- Enables inspection of data that applications are publishing, on any DDS domain.
- Can subscribe to select DDS *Topics* and display data samples it receives to the terminal.

rtiddsspy enables you to further narrow down the problem space by filtering data by *Topic*, setting Quality of Service (QoS) profiles, and monitoring specific Data-Distribution Service™ (DDS) domains, along with many other options.

Chapter 2

Usage

```
rtiddsspy
      [-help]
      [-version]
      [-domainId <domainId>]     ... defaults to 0
      [-domainTag <domainTag>]   ... defaults to ""
      [-index <NN>]              ... defaults to -1 (auto)
      [-appId <ID>]              ... defaults to a middleware-selected value
      [-verbosity <NN>]          ... can be 0..5
      [-peer <PEER>]             ... PEER format is NN\@TRANSPORT://ADDRESS
      [-discoveryTTL <NN>]       ... can be 0..255
      [-transport <MASK>]        ... defaults to DDS_TRANSPORTBUILTIN_MASK_
↪DEFAULT
      [-msgMaxSize <SIZE>]       ... defaults to -1 (transport default)
      [-shmRcvSize <SIZE>]       ... defaults to -1 (transport default)
      [-tcMaxSize <SIZE>]        ... defaults to 4096
      [-qosFile <file>]
      [-qosProfile <lib::prof>]

      [-hOutput]
      [-mode <USER|DISC>]
      [-deadline <SS>]         ... defaults to -1 (no deadline)
      [-timeFilter <SS>]         ... defaults to  0 (no filter)
      [-history <DEPTH>]       ... defaults to 8192
      [-partition <partition>]   ... defaults to *
      [-useFirstPublicationQos]
      [-showHandle]
      [-showSampleFlags]
      [-showSampleIdentity]
      [-typeRegex <REGEX>]       ... defaults to "*"
      [-topicRegex <REGEX>]      ... defaults to "*"
      [-typeWidth <WIDTH>]        ... can be 1..255
      [-topicWidth <WIDTH>]       ... can be 1..255
      [-printSample <PLAIN|COMPACT>]  ... default PLAIN
      [-showEntityName]
      [-showPartition]
```

Example:

```
rtiddsspy -domainId 3 -topicRegex "Alarm*"
```

To run *rtiddsspy*, like any DDS application, you must set the `NDDS_DISCOVERY_PEERS` environment variable that defines your DDS domain; otherwise you must specify the peers as command-line parameters. See How do I set my discovery peers, in the RTI Connext DDS Getting Started Guide.

To run *rtiddsspy* on a target system (not your host development platform), you must first select the target architecture. To do so, either:

- Set the environment variable `CONNEXTDDS_ARCH` to the name of the target architecture. (Do this for each command shell you will be using.)
- Or set the variable `connextdds_architecture` in the file `rticommon_config.[sh/bat]` to the name of the target architecture. (The file is `resource/scripts/rticommon_config.sh` on Linux® or macOS® systems, `resource/scripts/rticommon_config.bat` on Windows systems.) If the `CONNEXTDDS_ARCH` environment variable is set, the architecture in this file will be ignored.

Note: If you have more than one *DataWriter* for the same *Topic*, and these *DataWriters* have different settings for the Ownership QoS, then *rtiddsspy* will only receive (and thus report on) the samples from the first *DataWriter*.

2.1 Generic Options

-help Prints a help message and exits.

Example: `rtiddsspy -help`

-version Prints the version and exits.

Example: `rtiddsspy -version`

-domainId <NN> Sets the domain ID. The valid range is 0 to 100.

Example: `rtiddsspy -domainId 31`

-domainTag <TAG> Sets the domain tag. This is a string value with a maximum of 255 characters.

Example: `rtiddsspy -domainTag "ENG. DEPT"`

-index <NN> Sets the participantIndex. If participantIndex is not -1 (auto), it must be different than the one used by all other applications in the same computer and domainId. If this is not respected, *rtiddsspy* (or the application that starts last) will get an initialization error.

Example: `rtiddsspy -index 2`

-appId <ID> Sets the application ID. If unspecified, the system will pick one automatically. This option is rarely used.

Example: `rtiddsspy -appId 34556`

-verbosity <NN> Sets the verbosity level. The range is 0 to 5.

- 0 has minimal output and does not echo the fact that data is being sent or received.
- 1 prints the most relevant statuses, including the sending and receiving of data. This is the default.
- 2 prints a summary of the parameters being used and echoes more detailed status messages.

3-5 mostly affects the verbosity used by the internal DDS modules that implement *rtiddsspy*. The output is not always readable; its main purpose is to provide information that may be useful to RTI's support team.

Example: `rtiddsspy -verbosity 2`

-peer <PEER> Specifies a PEER to be used for discovery. Like any DDS application, it defaults to the setting of the environment variable `NDDS_DISCOVERY_PEERS` or a preconfigured multicast address if the environment is not set.

The format used for PEER is the same used for the `NDDS_DISCOVERY_PEERS` and is described in detail in Configuring the Peers List Used in Discovery. A brief summary follows:

The general format is: `NN\@TRANSPORT://ADDRESS` where:

- **ADDRESS** is an address (in name form or using the IP notation xxx.xxx.xxx.xxx). ADDRESS may be a multicast address. ADDRESS cannot be omitted if the `-peer` option is specified.
- **TRANSPORT** represents the kind of transport to use and NN is the maximum participantIndex expected at that location. NN can be omitted and defaults to '4'.
- Valid settings for **TRANSPORT** are 'udpv4' and 'shmem'. The default setting if the transport is omitted is 'udpv4'.

The `-peer` option may be repeated to specify multiple peers.

Example: `rtiddsspy -peer 10.10.1.192 -peer mars -peer 4\@pluto`

-discoveryTTL <TTL> Sets the TTL (time-to-live) used for multicast discovery. If not specified, it defaults to the builtin DDS default.

The valid range is 0 to 255. The value '0' limits multicast to the node itself (i.e., can only discover applications running on the same computer). The value '1' limits multicast discovery to computers on the same subnet. Settings greater than 1 generally indicate the maximum number of routers that may be traversed (although some routers may be configured differently).

Example: `rtiddsspy -discoveryTTL 16`

-transport <MASK> Specifies a bit-mask that sets the enabled builtin transports. If not specified, the default set of transports is used (UDPv4 + shmem).

The bit values are: `1=UDPv4, 2=shmem, 8=UDPv6`.

Example: `rtiddsspy -transport 1`

-msgMaxSize <SIZE> Configures the maximum message size allowed by the installed transports. This option is needed if you are using *rtiddsspy* to communicate with an application that has set these transport parameters to larger-than-default values.

Example: `rtiddsspy -msgMaxSize 1024`

-shmRcvSize <SIZE> Increases the shared memory receive-buffer size. This is needed if you are using *rtiddsspy* to communicate with an application that has set these transport parameters to larger-than-default values.

Example: `rtiddsspy -shmRcvSize 1024`

-tcMaxSize <SIZE> Configures the maximum size, in bytes, of a received type code.

Example: `rtiddsspy -tcMaxSize 1024`

-qosFile <file> Allows you to specify additional QoS XML settings using **url_profile**. For more information on the syntax, see How to Load XML-Specified QoS Settings, in the RTI Connext DDS Core Libraries User's Manual.

Example: `rtiddsspy -qosFile /home/user/QoSProfileFile.xml`

-qosProfile <lib::prof> Specifies the library name and profile name to be used.

Example: `rtiddsspy -qosProfile BuiltinQosLibExp:Generic.MinimalMemoryFootprint`

2.2 Spy-Specific Options

-hOutput Prints an explanation of the table in the output and then exits. (These explanations are also described in *Understanding the Output*.)

Example: `rtiddsspy -hOutput`

-mode <USER|DISC> Chooses between printing discovery data or user data. See *Discovery vs. User Modes*.

- USER: Only prints user data samples. Skips discovery events. Also sets `-printSample` by default.
- DISC: Only prints discovery events. Skips user data samples.

By default, if you do not specify a mode, *rtiddsspy* prints both discovery events and user data samples.

Example: `rtiddsspy -mode USER`

-deadline <SS> Sets the requested DEADLINE QoS for the subscriptions made by *rtiddsspy*.

Note that the deadline period should be equal to or greater than the publisher's deadline period. If not, the QoS will be incompatible. If the QoS is incompatible, *rtiddsspy* will not receive updates from that publisher.

Each time a deadline is detected, *rtiddsspy* will print a message that indicates the number of deadlines received so far.

Example: `rtiddsspy -deadline 3.5`

-timeFilter <SS> Sets the TIME_BASED_FILTER QoS for the subscriptions made by *rtiddsspy*. This QoS causes DDS to filter out messages that are published at a rate faster than what the filter duration permits. For example, if the filter duration is 10 seconds, messages will be printed no faster than once every 10 seconds.

Example: `rtiddsspy -timeFilter 10.0`

-history <DEPTH> Sets the HISTORY depth QoS for the subscriptions made by *rtiddsspy*.

This option may be relevant if the publisher has batching turned on, or if you are using the `-useFirstPublicationQos` option, which causes a reliable or durable subscription to be created.

Example: `rtiddsspy -history 1`

-partition <PARTITION> Adds a partition to the list of partition names. This option can be repeated to indicate multiple partitions. The first occurrence of this parameter resets the default partition list.

Default: * (unless overwritten with `-qosProfile` option)

Example: `rtiddsspy -partition A`

-useFirstPublicationQos Sets the RELIABILITY and DURABILITY QoS of the subscription based on the first discovered publication of that topic.

See also `-history` option.

Example: `rtiddsspy -useFirstPublicationQos`

-showHandle Prints additional information about each sample received. The additional information is the **instance_handle** field in the SampleHeader, which can be used to distinguish among multiple instances of data objects published under the same *Topic* and type names.

Samples that share the *Topic* and type names and have the same **instance_handle** value, represent value updates to the same data object. Samples that share the *Topic* and type names but have different **instance_handle** values, represent different data objects.

This option causes *rtiddsspy* to print an explanation of updates to the values of different data objects.

Example: `rtiddsspy -showHandle`

-showSampleFlags Adds one field with the active flags associated with each sample.

Example: `rtiddsspy -showSampleFlags`

-showSampleIdentity Adds two fields with the sample identity and related sample identity associated with each sample.

Example: `rtiddsspy -showSampleIdentity`

-typeRegex <REGEX> Subscribes only to types that match the REGEX regular expression. The regular expression follows the UNIX style fnmatch syntax.

When typing a regular expression to a command-line shell, some symbols may need to be escaped to avoid interpretation by the shell. In general, it is safest to include the expression in double quotes.

This option may be repeated to specify multiple topic expressions.

Example: `rtiddsspy -typeRegex "SensorArray*"`

-topicRegex <REGEX> Subscribes only to *Topics* that match the REGEX regular expression. The regular expression follows the UNIX style fnmatch syntax.

When typing a regular expression to a command-line shell, some symbols may need to be escaped to avoid interpretation by the shell. In general, it is safest to include the expression in double quotes.

This option may be repeated to specify multiple *Topic* expressions.

Example: `rtiddsspy -topicRegex "Alarm*"`

-typeWidth <WIDTH> Sets the maximum width of the type name field in the output. For example, if you don't have room to display the whole name, you could reduce the width. The valid range is 1 to 255.

Example: `rtiddsspy -typeWidth 123`

-topicWidth <WIDTH> Sets the maximum width of the *Topic* name field in the output. For example, if you don't have room to display the whole name, you could reduce the width. The valid range is 1 to 255.

Example: `rtiddsspy -topicWidth 123`

-printSample <PLAIN|COMPACT> Prints the value of the received samples. See *Sample Display Options*.

- PLAIN: Default. Pretty-prints sample information for best readability.
- COMPACT: Prints sample information in a single line using a JSON format.

Example: `rtiddsspy -printSample COMPACT`

-showEntityName Displays the entity name each time a sample is received.

Example: `rtiddsspy -showEntityName`

-showPartition Displays the partitions to which an entity belongs each time a new entity is discovered and sample is received.

See PARTITION QosPolicy, in the RTI Connext DDS Core Libraries User's Manual for more information.

Example: `rtiddsspy -showPartition`

2.3 QoS Settings

rtiddsspy is configured to discover as many entities as possible. To do so, an internal profile is defined, called `InternalSpyLibrary::InternalSpyProfile`. This is the default profile, unless a profile called `DefaultSpyLibrary::DefaultSpyProfile` is found.

You can use the command-line option `-qosProfile` to tell *rtiddsspy* to use a specific `lib::profile` instead of `DefaultSpyLibrary::DefaultSpyProfile`.

Like all the other DDS applications, *rtiddsspy* loads all the profiles specified using the environment variable `NDDS_QOS_PROFILES` or the file named `USER_QOS_PROFILES` found in the current working directory.

The QoS settings used internally are available in the file `RTIDDSSPY_QOS_PROFILES.example.xml`.

2.4 Usage Examples for Linux and Windows Systems

On Linux, Windows, and other operating systems that have a shell, the syntax matches the regular commands available in the shell. (In the examples below, the string `shell prompt>` represents the prompt that the shell prints and is not part of the command that must be typed.)

```
shell prompt>  rtiddsspy -domainId 3
shell prompt>  rtiddsspy -domainId 5 -topicRegex "Alarm*"
shell prompt>  rtiddsspy -help
```

2.5 Usage Examples for VxWorks Systems

On VxWorks® systems, the libraries `libnddscore.so`, `libnddsc.so`, and `libnddscpp.so` must first be loaded.

The *rtiddsspy* command must be typed into the VxWorks shell (either an rlogin shell, a target-server shell, or the serial line prompt). The arguments are passed embedded into a single string, but otherwise have the same syntax as Windows systems. On Linux, Windows, and other operating systems that have a shell, the syntax matches the regular commands available in the shell. (In the examples below, the string `vxworks prompt>` represents the prompt that the shell prints and is not part of the command that must be typed.)

```
rtiddsspy "[<options>]"
Example: rtiddsspy "-domainId 3 -topicRegex Alarm*"
```

rtiddsspy requires about 25 kB of stack. If the stack size of the shell from which it is invoked is not large enough, use `taskSpawn`:

```
taskSpawn <name>, <priority>, <taskspawn options>, <stack size in bytes>,␣
↪rtiddsspy, "[<options>]"
```

The options use the same syntax as above. For example:

```
taskSpawn "rtiddsspy", 100, 0x8, 50000, rtiddsspy, "-domainId 3 -topicRegex␣
↪Alarm*"
```

Chapter 3

Understanding the Output

rtiddsspy's output is formatted as a list of lines, where each line is independent and describes a sample received.

```
RTI Connext DDS Spy
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

16:12:49 New writer        from 192.168.43.223  : topic="rti/distlog" type=
↪"com::rti::dl::LogMessage"
16:12:49 New writer        from 192.168.43.223  : topic="rti/distlog/
↪administration/state" type="com::rti::dl::admin::State"
16:12:49 New writer        from 192.168.43.223  : topic="rti/distlog/
↪administration/command_response" type="com::rti::dl::admin::CommandResponse"
12:12:29 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
16:12:49 New reader        from 192.168.43.223  : topic="rti/distlog/
↪administration/command_request" type="com::rti::dl::admin::CommandRequest
↪"
09:34:35 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
09:34:35 Modified instance from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"

---- Statistics ----
Discovered 4 DataWriters and 1 DataReaders
Received samples (Data, Dispose, NoWriters):
    0, 0, 0          (Topic="rti/distlog"  Type="com::rti::dl::LogMessage")
    0, 0, 0          (Topic="rti/distlog/administration/state"  Type=
↪"com::rti::dl::admin::State")
    0, 0, 0          (Topic="rti/distlog/administration/command_response" ␣
↪Type="com::rti::dl::admin::CommandResponse")
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

The descriptions of output described in the following sections are the same as provided when you use the `-hOutput` option (described in *Spy-Specific Options*).

3.1 Default Output

In the default output, you can find the following information per line:

source_timestamp Timestamp that appears in the SampleInfo associated with each sample. This timestamp represents the source-time when the sample was generated. It is printed in Coordinated Universal Time (UTC).

For example: `14:59:16`.

A timestamp of `xx:xx:xx` indicates that the sample was not valid. For example, it could have been a dispose sample.

Info Description of the received sample. The value distinguishes between discovery and user traffic.

For example: `New writer`.

- **"New writer"/"New reader":** Indicates discovery traffic. It means that *rtiddsspy* found a new writer/reader in another application.
- **"Updated writer"/"Updated reader":** Indicates discovery traffic.

  It means that the entity was explicitly updated by an application by changing any entity configuration parameter such as the partition, lifespan, or content-filtered topic expression.
- **"Deleted writer"/"Deleted reader":** Indicates discovery traffic.

  It means that the entity was explicitly deleted by an application or has otherwise lost liveliness.
- **"New data" indicates user traffic.** For keyed data: Indicates a sample was received for a new instance that *rtiddsspy* has never seen before.

  For unkeyed data: Indicates a sample was received.
- **"Modified instance" indicates user traffic.** Indicates a modified instance that *rtiddsspy* has seen before.
- **"Disposed instance" indicates user traffic.** Indicates that an instance was disposed.
- **"No writers" indicates user traffic.** Indicates that an instance has no more registered writers.

Src IP IPv4 address of the writer that sent the sample. This value is the first IPv4 element of the **unicast_locators** of the *DataWriter* or its *DomainParticipant*. If no such element exists or it cannot be parsed, then `unknown ip` is displayed.

For example: `from 192.168.43.223`.

Topic *Topic* name. For discovery messages, this refers to the *Topic* that has been discovered.

For example: `topic="Circle"`.

Type Type name. For discovery messages, this refers to the type of the *Topic* that has been discovered.

For example: `type="ShapeType"`.

Statistics *rtiddsspy* prints how many endpoints have been discovered and how many samples have been received, by *Topic*.

In the following example, *rtiddsspy* has discovered 4 *DataWriters* and 1 *DataReader*. *rtiddsspy* has received 2 samples from the *Topic* `"Circle"` and 0 samples from the rest of the *Topics*.

```
---- Statistics ----
Discovered 4 DataWriters and 1 DataReaders
Received samples (Data, Dispose, NoWriters):
    0, 0, 0          (Topic="rti/distlog"  Type=
↪"com::rti::dl::LogMessage")
    0, 0, 0          (Topic="rti/distlog/administration/state" ␣
↪Type="com::rti::dl::admin::State")
    0, 0, 0          (Topic="rti/distlog/administration/command_
↪response"  Type="com::rti::dl::admin::CommandResponse")
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

3.2 Additional Output

The following fields do not appear by default, they need to be requested by command-line arguments:

Object GUID/Key To enable this information, use `-showHandle`.

This field contains the 12-byte instance handle that appears in the SampleInfo associated with each sample.

Note that the information is only displayed for *Topics* that have a key. Note also that for simple key formats that can fit in 12 bytes, it will exactly match the key.

For example:

- For discovery data: `guid="0x0101888C,0x46C9EF71, 0xFA7ED5AE:0x80004202"`.
- For user data: `key="c317c2ca.8e3f3618.ee0e16f1.86e8f9de"`.

Sample Identity To enable this information, use `-showSampleIdentity`.

This field contains the original publication virtual GUID and the original publication virtual sequence number that appears in the SampleInfo associated with each sample.

For example: `virtual_guid="0x0101888C,0x46C9EF71, 0xFA7ED5AE:0x80004202" virtual_sn="(0, 6140280)"`.

Related Sample Identity To enable this information, use `-showSampleIdentity`.

This field contains the related original publication virtual GUID and the related original publication virtual sequence number that appears in the SampleInfo associated with each sample.

For example: `related_virtual_guid="0x0101888C,0x46C9EF71, 0xFA7ED5AE:0x80004202" related_virtual_sn="(0, 6140280)"`.

Flags To enable this information, use `-showSampleFlags`.

This field shows the active flags associated with each sample.

For example: `flags="LR"`.

The possible flags are:

- `R`: the sample has been redelivered by *RTI Queuing Service*.
- `I`: a response sample is not the last sample for a given request. This bit is usually set by Connext Repliers sending multiple responses for a request.
- `P`: a sample must be broadcast by one *RTI Queuing Service* replica to other replicas.
- `L`: a response sample is the last sample in a SharedReaderQueue for a QueueConsumer *DataReader*.

Entity Name To enable this information, use `-showEntityName`.

This field contains the name of the entity discovered or the name of the *DataWriter* sending data. For example: `name="roleName:testDataWriter"`.

By default, the entity name is printed each time an entity is discovered. For example: `16:12:49 New writer from 192.168.43.223 : topic="rti/distlog" type="com::rti::dl::LogMessage"`

By default, the *DataWriter* name is not printed each time it sends data. For example: `09:34:35 New data from 192.168.43.223 : topic="Circle" type="ShapeType"`

Partition To enable this information, use `-showPartition`.

This field contains the partition of the entity discovered or the partition of the *DataWriter* sending data.

For example: `partition="A,B,C"`.

For information on partitions, see PARTITION QosPolicy, in the RTI Connext DDS Core Libraries User's Manual.

Sample There are two options to display the sample:

- `-printSample PLAIN`: This is the default option. It pretty-prints sample information for best readability.

```
12:35:24 New data          from 192.168.43.223  :␣
↪topic="Circle" type="ShapeType"
color: "BLUE"
x: 53
y: 190
shapesize: 30
fillKind: SOLID_FILL
angle: 0
```

- `-printSample COMPACT`: This option prints sample information in a single line using a JSON format.

```
12:38:15 New data          from 192.168.43.223  :␣
↪topic="Circle" type="ShapeType" sample={"color":"BLUE
↪","x":202,"y":175,"shapesize":30,"fillKind":"SOLID_
↪FILL","angle":0}
```

Chapter 4

Examples

4.1 Default Behavior

```
./rtiddsspy

RTI Connext DDS Spy
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

12:12:29 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
12:32:18 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
12:32:18 Modified instance from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"

---- Statistics ----
Discovered 1 DataWriters and 0 DataReaders
Received samples (Data, Dispose, NoWriters):
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

4.2 Sample Display Options

There are two options to display the sample information.

4.2.1 Plain (Default)

This is the default option. It pretty-prints sample information for best readability.

```
./rtiddsspy -printSample PLAIN

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

12:12:29 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
12:35:24 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
color: "BLUE"
x: 53
y: 190
shapesize: 30
fillKind: SOLID_FILL
angle: 0

12:35:24 Modified instance from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
color: "BLUE"
x: 54
y: 195
shapesize: 30
fillKind: SOLID_FILL
angle: 0

---- Statistics ----
Discovered 1 DataWriters and 0 DataReaders
Received samples (Data, Dispose, NoWriters):
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

4.2.2 Compact

This option prints sample information in a single line using a JSON format.

```
./rtiddsspy -printSample COMPACT

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

```

(continues on next page)

(continued from previous page)

```
12:12:29 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
12:38:15 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" sample={"color":"BLUE","x":202,"y":175,"shapesize":30,"fillKind
↪":"SOLID_FILL","angle":0}
12:38:15 Modified instance from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" sample={"color":"BLUE","x":203,"y":180,"shapesize":30,"fillKind
↪":"SOLID_FILL","angle":0}

---- Statistics ----
Discovered 1 DataWriters and 0 DataReaders
Received samples (Data, Dispose, NoWriters):
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

4.3 Discovery vs. User Modes

By default, *rtiddsspy* prints both discovery and user data together; however, you can choose to print either discovery or user data.

4.3.1 Discovery Mode

Use `-mode DISC` to print just discovery data and leave out user data. This mode forces the `-showEntityName`, `-showHandle`, and `-showPartition` arguments.

```
./rtiddsspy -mode DISC

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

13:26:17 New reader        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" guid="0x0101CEB8,0x3426926B,0xA577212D:0x80000004" partition=
↪"A, B, C" name="roleName:testDataReader"
13:26:15 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" guid="0x0101A438,0xBAE7B797,0x5443C840:0x80000003" partition=
↪"A, E, I" name="roleName:testDataWriter"
13:29:22 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" guid="0x0101B1F3,0xE6C768BE,0xFB33BCF3:0x80000003" partition=
↪"D, F, G" name="roleName:testDataWriter"


---- Statistics ----
Discovered 2 DataWriters and 1 DataReaders
```

4.3.2 User Mode

Use `-mode USER` to print user data and leave out discovery data. This mode forces the `-printSample` argument.

```
./rtiddsspy -mode USER

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

15:14:37 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
color: "BLUE"
x: 53
y: 190
shapesize: 30
fillKind: SOLID_FILL
angle: 0

15:14:38 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
color: "BLUE"
x: 54
y: 195
shapesize: 30
fillKind: SOLID_FILL
angle: 0

---- Statistics ----
Discovered 2 DataWriters and 1 DataReaders
Received samples (Data, Dispose, NoWriters):
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

4.4 Additional Display Options

This example uses arguments that print all of the extra information related to discovery and user data.

```
./rtiddsspy -showEntityName -showPartition -showHandle -showSampleIdentity -
↪showSampleFlags -printSample COMPACT

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

13:26:17 New reader        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" guid="0x0101CEB8,0x3426926B,0xA577212D:0x80000004" name=
↪"roleName:testDataReader" partition="A, B, C"
13:26:15 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" guid="0x0101A438,0xBAE7B797,0x5443C840:0x80000003" name=
↪"roleName:testDataWriter" partition="A, E, I"
```

(continues on next page)

(continued from previous page)

```
15:41:09 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" virtual_guid="0x0101A438,0xBAE7B797,0x5443C840:0x80000003"␣
↪virtual_sn="(0, 8093)" name="roleName:testDataWriter" partition="A, E, I
↪" sample={"color":"BLUE","x":202,"y":175,"shapesize":30,"fillKind":"SOLID_
↪FILL","angle":0}
15:41:10 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType" virtual_guid="0x0101A438,0xBAE7B797,0x5443C840:0x80000003"␣
↪virtual_sn="(0, 8094)" name="roleName:testDataWriter" partition="A, E, I
↪" sample={"color":"BLUE","x":203,"y":180,"shapesize":30,"fillKind":"SOLID_
↪FILL","angle":0}

---- Statistics ----
Discovered 1 DataWriters and 1 DataReaders
Received samples (Data, Dispose, NoWriters):
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

4.5 Filter by Topic and Type

This example uses arguments to filter discovery and user data by *Topic* and type.

```
./rtiddsspy -typeRegex "Shape*" -topicRegex "Circ*"

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

13:26:17 New reader        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
13:26:15 New writer        from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
16:24:16 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"
16:24:17 New data          from 192.168.43.223  : topic="Circle" type=
↪"ShapeType"


---- Statistics ----
Discovered 1 DataWriters and 1 DataReaders
Received samples (Data, Dispose, NoWriters):
    2, 0, 0          (Topic="Circle"  Type="ShapeType")
```

4.6 Partitions

4.6.1 Display All Partitions

This example displays the partitions to which an entity belongs each time a new entity is discovered and a sample is received. (See PARTITION QosPolicy, in the RTI Connext DDS Core Libraries User's Manual for more information on partitions.)

```
./rtiddsspy -showPartition

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

13:29:22 New writer        from 192.168.43.223  : topic="Example test" type=
↪"test" partition="D"
13:26:15 New writer        from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"
13:26:17 New reader        from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"
13:29:42 New data          from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"
13:29:43 New data          from 192.168.43.223  : topic="Example test" type=
↪"test" partition="D"
13:29:43 New data          from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"
13:29:44 New data          from 192.168.43.223  : topic="Example test" type=
↪"test" partition="D"


---- Statistics ----
Discovered 2 DataWriters and 1 DataReaders
Received samples (Data, Dispose, NoWriters):
    4, 0, 0          (Topic="Example test"  Type="test")
```

4.6.2 Display Select Partitions

This example shows how to display data only for specified partitions. By using `-partition A`, we only receive data from entities that belong to partition `A`.

Although *rtiddsspy* first displays all discovered entities (`New writer`, `New reader`), regardless of partition, any data received (`New data`) is shown only for the specified partition.

```
./rtiddsspy -showPartition -partition A

RTI Connext DDS Spy built with DDS version: |CONNEXT_currentVersion|
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rtiddsspy is listening for data, press CTRL+C to stop it.

13:29:22 New writer        from 192.168.43.223  : topic="Example test" type=
↪"test" partition="D"
```

(continues on next page)

(continued from previous page)

```
13:26:15 New writer         from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"
13:26:17 New reader         from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"
13:30:24 New data           from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"
13:30:25 New data           from 192.168.43.223  : topic="Example test" type=
↪"test" partition="A"


---- Statistics ----
Discovered 2 DataWriters and 1 DataReaders
Received samples (Data, Dispose, NoWriters):
    2, 0, 0          (Topic="Example test"  Type="test")
```

Chapter 5

Copyrights and Notices

Trademarks

Copy and Use Restrictions

Notices

Deprecations and Removals

Any deprecations or removals noted in this document serve as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220 and/or any other agreements by and between RTI and customer regarding maintenance and support of RTI's software.

Deprecated means that the item is still supported in the release, but will be removed in a future release. *Removed* means that the item is discontinued or no longer supported. By specifying that an item is deprecated in a release, RTI hereby provides customer notice that RTI reserves the right after one year from the date of such release and, with or without further notice, to immediately terminate maintenance (including without limitation, providing updates and upgrades) for the item, and no longer support the item, in a future release.

Early Access Software

"Real-Time Innovations, Inc. ("RTI") licenses this Early Access release software ("Software") to you subject to your agreement to all of the following conditions:

(1) you may reproduce and execute the Software only for your internal business purposes, solely with other RTI software licensed to you by RTI under applicable agreements by and between you and RTI, and solely in a non-production environment;

(2) you acknowledge that the Software has not gone through all of RTI's standard commercial testing, and is not maintained by RTI's support team;

(3) the Software is provided to you on an "AS IS" basis, and RTI disclaims, to the maximum extent permitted by applicable law, all express and implied representations, warranties and guarantees, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, satisfactory quality, and non-infringement of third party rights;

(4) any such suggestions or ideas you provide regarding the Software (collectively , "Feedback"), may be used and exploited in any and every way by RTI (including without limitation, by granting sublicenses), on a non-exclusive, perpetual, irrevocable, transferable, and worldwide basis, without any compensation, without any obligation to report on such use, and without any other restriction or obligation to you; and

(5) TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL RTI BE LIABLE TO YOU FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY OR PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR FOR LOST PROFITS, LOST DATA, LOST REPUTATION, OR COST OF COVER, REGARDLESS OF THE FORM OF ACTION WHETHER IN CONTRACT, TORT (INCLUDING WITHOUT LIMITATION, NEGLIGENCE), STRICT PRODUCT LIABILITY OR OTHERWISE, WHETHER ARISING OUT OF OR RELATING TO THE USE OR INABILITY TO USE THE SOFTWARE, EVEN IF RTI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES."

Technical Support Real-Time Innovations, Inc. 232 E. Java Drive Sunnyvale, CA 94089 Phone: (408) 990-7444 Email: support@rti.com Website: https://support.rti.com/