

RTI Limited Bandwidth Endpoint Discovery Plugin

User's Manual

Version 6.0.0



© 2019 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
February 2019.

Trademarks

Real-Time Innovations, RTI, NDDS, RTI Data Distribution Service, Connex, Micro DDS, the RTI logo, IRTI and the phrase, “Your Systems. Working as one,” are registered trademarks, trademarks or service marks of Real-Time Innovations, Inc. All other trademarks belong to their respective owners.

Copy and Use Restrictions

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form (including electronic, mechanical, photocopy, and facsimile) without the prior written permission of Real-Time Innovations, Inc. The software described in this document is furnished under and subject to the RTI software license agreement. The software may be used or copied only under the terms of the license agreement.

This is an independent publication and is neither affiliated with, nor authorized, sponsored, or approved by, Microsoft Corporation.

The security features of this product include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

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/>

Contents

Chapter 1 Introduction

1.1 What is Discovery?	1
1.2 Configuring Properties in XML	2

Chapter 2 Paths Mentioned in Documentation 3

Chapter 3 Limited Bandwidth Endpoint Discovery Plugin

3.1 Creating the LBED Plugin Configuration File	5
3.2 Configuring LBED Plugin in Connex DDS	16
3.3 Optimizing the Plugin	18

Chapter 1 Introduction

The *RTI® Limited Bandwidth Endpoint Discovery (LBED) Plugin* reduces discovery time and network traffic by obtaining information about the endpoints from an XML file (instead of from the normal discovery process, which requires the information to be sent dynamically over the network). All the endpoints must be known ahead of time and described in an XML file.

1.1 What is Discovery?

Discovery is the behind-the-scenes way in which *RTI Connex® DDS* objects (*DomainParticipants*, *DataWriters*, and *DataReaders*) find out about each other. Each *DomainParticipant* maintains a database of information about all the active *DataReaders* and *DataWriters* in the same domain. This database is what makes it possible for *DataWriters* and *DataReaders* to communicate. To create and refresh the database, each application follows a common discovery process.

The default discovery mechanism in *Connex DDS* is the one described in the DDS specification and is known as Simple Discovery Protocol, which includes two phases: Simple Participant Discovery and Simple Endpoint Discovery. The goal of these two phases is to build, for each *DomainParticipant*, a complete picture of all the entities that belong to the remote participants in its peers list, which is a list of nodes with which a participant may communicate.

During the Simple Participant Discovery phase, *DomainParticipants* learn about each other. The *DomainParticipant's* details are communicated to all other *DomainParticipants* in the same domain by sending participant declaration messages, also known as participant DATA submessages.

During the Simple Endpoint Discovery phase, *Connex DDS* matches *DataWriters* and *DataReaders*. Information about each application's *DataReaders* and *DataWriters* is exchanged by sending publication/subscription declarations in DATA submessages, which we will refer to as publication DATAs and subscription DATAs. The Simple Endpoint Discovery phase uses reliable communication.

With Limited Bandwidth Endpoint Discovery (LBED), the Simple Participant Discovery phase still occurs, but the information normally gathered dynamically by Simple Endpoint Discovery is instead configured statically in an XML file.

1.2 Configuring Properties in XML

Connex DDS provides a mechanism to dynamically load an external transport from an XML QoS profile, like the file generated by *rtiddsgen* (**USER_QOS_PROFILES.xml**). The Property QoS policy is used to achieve this purpose.

The Property QoS policy stores name/value (string) pairs that can be used to configure certain parameters of *Connex DDS* that are not exposed through formal QoS policies. *Connex DDS* uses this mechanism to configure external transports.

Syntax for Setting the Property QoS Policy in an XML QoS profile:

```
<qos_library name="Property_Library">
  <qos_profile name="Property_Profile">
    <participant_qos>
      ...
      <property>
        <value>
          <element>
            <name>Property1</name>
            <value>example</value>
          </element>
          <element>
            <name>Property2</name>
            <value>example</value>
          </element>
          ...
        </value>
      </property>
      ...
    </participant_qos>
  </qos_profile>
</qos_library>
```

For more general information, see the chapter on *Configuring QoS with XML* in the *RTI Connex DDS Core Libraries User's Manual*.

For specific information on setting properties for the LBED plugin, see [Chapter 3 Limited Bandwidth Endpoint Discovery Plugin on page 5](#).

Chapter 2 Paths Mentioned in Documentation

The documentation refers to:

- **<NDDSHOME>**

This refers to the installation directory for *RTI® Connex® DDS*. The default installation paths are:

- Mac® OS X® systems:
/Applications/rti_connex_dds-6.0.0
- UNIX-based systems, non-*root* user:
/home/<your user name>/rti_connex_dds-6.0.0
- UNIX-based systems, *root* user:
/opt/rti_connex_dds-6.0.0
- Windows® systems, user without Administrator privileges:
<your home directory>\rti_connex_dds-6.0.0
- Windows systems, user with Administrator privileges:
C:\Program Files\rti_connex_dds-6.0.0

You may also see \$NDDSHOME or %NDDSHOME%, which refers to an environment variable set to the installation path.

Wherever you see <NDDSHOME> used in a path, replace it with your installation path.

Note for Windows Users: When using a command prompt to enter a command that includes the path **C:\Program Files** (or any directory name that has a space), enclose the path in quotation marks. For example:

```
"C:\Program Files\rtdi_connext_dds-6.0.0\bin\rtiddsgen"
```

Or if you have defined the `NDDSHOME` environment variable:

```
"%NDDSHOME%\bin\rtiddsgen"
```

- **<path to examples>**

By default, examples are copied into your home directory the first time you run *RTI Launcher* or any script in `<NDDSHOME>/bin`. This document refers to the location of the copied examples as `<path to examples>`.

Wherever you see `<path to examples>`, replace it with the appropriate path.

Default path to the examples:

- Mac OS X systems: `/Users/<your user name>/rtdi_workspace/6.0.0/examples`
- UNIX-based systems: `/home/<your user name>/rtdi_workspace/6.0.0/examples`
- Windows systems: `<your Windows documents folder>\rtdi_workspace\6.0.0\examples`

Where 'your Windows documents folder' depends on your version of Windows. For example, on Windows 10, the folder is `C:\Users\<your user name>\Documents`.

Note: You can specify a different location for `rtdi_workspace`. You can also specify that you do not want the examples copied to the workspace. For details, see *Controlling Location for RTI Workspace and Copying of Examples* in the *RTI Connext DDS Core Libraries Getting Started Guide*.

Chapter 3 Limited Bandwidth Endpoint Discovery Plugin

Limited Bandwidth Endpoint Discovery (LBED) is achieved with a file-based plugin. Information about the endpoints is obtained from an XML file instead of being sent dynamically over the network. This method can reduce discovery time and reduce network traffic. However, for LBED to work, all the endpoints must be known ahead of time and described in an XML file.

This chapter describes how to configure the LBED Plugin and set up your *Connex DDS* application to use it.

You will need two XML files, one for the discovery plugin (see [3.1 Creating the LBED Plugin Configuration File below](#)) and one for *Connex DDS* ([3.2 Configuring LBED Plugin in Connex DDS on page 16](#)).

You must link with the *dynamic* version of the *Connex DDS* libraries. See the *RTI Connex DDS Core Libraries Platform Notes* for details.

3.1 Creating the LBED Plugin Configuration File

To use LBED, you need an XML file that describes all the remote participants and their endpoints. These remote endpoints must be configured exactly the same as their original QoS properties.

You will specify the name of this file when you configure the plugin in the QoS Profiles XML file (`USER_QOS_PROFILES.xml`) described in [3.2 Configuring LBED Plugin in Connex DDS on page 16](#); see `dds.discovery.endpoint.<string>.config_file`

The main structure of the XML file is:

```
<LBEDiscoveryPluginProfile>
  <participant name="myParticipant">
    <datareader>
      ...
    </datareader>
    ...
  </participant>
  ...
</LBEDiscoveryPluginProfile>
```



```

    <datawriter>
      ...
    </datawriter>
  </participant>
</LBEDiscoveryPluginProfile>

```

The supported configuration options for *DataReaders* and *DataWriters* are described in [Table 3.1 Configuration Properties for LBED Plugin](#). The descriptions for many of these options also point out related *Connex DDS* documentation. For example, “See documentation on the Deadline QoS policy” means you should see that section in the *RTI Connex DDS Core Libraries User's Manual* or API Reference HTML documentation. These options can be set for *DataReaders* and *DataWriters*, unless otherwise noted in [Table 3.1 Configuration Properties for LBED Plugin](#).

The following example shows an example LBED plugin configuration file. You can find this file in **<path to examples>/connex_dds/c/limited_bandwidth_plugins/lbediscovery/LBEDiscoveryPluginExamplePublisher.xml**:

```

<?xml version="1.0"?>
<LBEDiscoveryPluginProfile xmlns:xsi=
  "http://www.w3.org/2001/XMLSchema-instance">
  <participant name="myParticipant2">
    <datawriter>
      <rtps_object_id>200</rtps_object_id>
      <topic_name>Hello LBEDiscovery</topic_name>
      <type_name>DDS::String</type_name>
      <topic_keyed>>false</topic_keyed>
      <durability>
        <kind>VOLATILE_DURABILITY_QOS</kind>
        <direct_communication>>true</direct_communication>
      </durability>
      <destination_order>
        <kind>
          BY_RECEPTION_TIMESTAMP_DESTINATIONORDER_QOS
        </kind>
        <source_timestamp_tolerance>
          <sec>30</sec>
          <nanosec>0</nanosec>
        </source_timestamp_tolerance>
      </destination_order>
      <presentation>
        <access_scope>
          INSTANCE_PRESENTATION_QOS
        </access_scope>
        <coherent_access>>false</coherent_access>
        <ordered_access>>false</ordered_access>
      </presentation>
      <deadline>
        <period>
          <sec>DURATION_INFINITE_SEC</sec>
          <nanosec>DURATION_INFINITE_NSEC</nanosec>
        </period>
      </deadline>
    </datawriter>
  </participant>
</LBEDiscoveryPluginProfile>

```

```

<latency_budget>
  <duration>
    <sec>0</sec>
    <nanosec>0</nanosec>
  </duration>
</latency_budget>
<liveliness>
  <kind>AUTOMATIC_LIVELINESS_QOS</kind>
  <lease_duration>
    <sec>DURATION_INFINITE_SEC</sec>
    <nanosec>DURATION_INFINITE_NSEC</nanosec>
  </lease_duration>
</liveliness>
<reliability>
  <kind>RELIABLE_RELIABILITY_QOS</kind>
  <max_blocking_time>
    <sec>0</sec>
    <nanosec>100000000</nanosec>
  </max_blocking_time>
  <acknowledgment_kind>
    PROTOCOL_ACKNOWLEDGMENT_MODE
  </acknowledgment_kind>
</reliability>
<ownership>
  <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
</ownership>
<ownership_strength>
  <value>0</value>
</ownership_strength>
<lifespan>
  <duration>
    <sec>DURATION_INFINITE_SEC</sec>
    <nanosec>DURATION_INFINITE_NSEC</nanosec>
  </duration>
</lifespan>
<durability_service>
  <service_cleanup_delay>
    <sec>0</sec>
    <nanosec>0</nanosec>
  </service_cleanup_delay>
  <history_kind>
    KEEP_LAST_HISTORY_QOS
  </history_kind>
  <history_depth>1</history_depth>
  <max_samples>LENGTH_UNLIMITED</max_samples>
  <max_instances>LENGTH_UNLIMITED</max_instances>
  <max_samples_per_instance>
    LENGTH_UNLIMITED
  </max_samples_per_instance>
</durability_service>
<service>NO_SERVICE</service>
</datawriter>
</participant>
</LBEDiscoveryPluginProfile>

```

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
deadline	<p>Maximum time between data samples. See documentation on the Deadline QoS policy.</p> <p>Schema:</p> <pre><deadline> <period> <sec>[number]</sec> <nanosec>[number]</nanosec> </period> </deadline></pre> <p>Example:</p> <pre><deadline> <period> <sec>DURATION_INFINITE_SEC</sec> <nanosec>DURATION_INFINITE_NSEC</nanosec> </period> </deadline></pre>
destination_order	<p>Controls how <i>Connext DDS</i> will deal with data sent by multiple DataWriters for the same topic. See documentation on the DestinationOrder QoS policy.</p> <p>Schema:</p> <pre><destination_order> <kind> [BY_RECEPTION_TIMESTAMP_DESTINATIONORDER_QOS BY_SOURCE_TIMESTAMP_DESTINATIONORDER_QOS] </kind> <source_timestamp_tolerance> <sec>[number]</sec> <nanosec>[number]</nanosec> </source_timestamp_tolerance> </destination_order></pre> <p>Example:</p> <pre><destination_order> <kind>BY_RECEPTION_TIMESTAMP_DESTINATIONORDER_QOS</kind> <source_timestamp_tolerance> <sec>30</sec> <nanosec>0</nanosec> </source_timestamp_tolerance> </destination_order></pre>
disable_positive_acks	<p>Whether or not to disable positive ACKs. See documentation on the DataWriterProtocol and DataReaderProtocol QoS policies.</p> <p>Schema:</p> <pre><protocol> <disable_positive_acks> [true false] </disable_positive_acks> </protocol></pre> <p>Example:</p> <pre><protocol> <disable_positive_acks>true</disable_positive_acks> </protocol></pre>

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
durability	<p>Specifies if <i>Connex DDS</i> will store and deliver previously published data to new/late-joining <i>DataReaders</i>. See documentation on the Durability QoS policy.</p> <p>Schema:</p> <pre><durability> <kind> [VOLATILE_DURABILITY_QOS TRANSIENT_LOCAL_DURABILITY_QOS TRANSIENT_DURABILITY_QOS PERSISTENT_DURABILITY_QOS] </kind> <direct_communication> [true false] </direct_communication> </durability></pre> <p>Example:</p> <pre><durability> <kind>VOLATILE_DURABILITY_QOS</kind> <direct_communication>true</direct_communication> </durability></pre>
durability_service	<p>Various settings to configure the external <i>Persistence Service</i> used by <i>Connex DDS</i> for <i>DataWriters</i> with a Durability setting of persistent. See documentation on the Durability QoS policy.</p> <p>Schema:</p> <pre><durability_service> <service_cleanup_delay> <sec>[number]</sec> <nanosec>[number]</nanosec> </service_cleanup_delay> <history_kind> [KEEP_LAST_HISTORY_QOS KEEP_ALL_HISTORY_QOS] </history_kind> <history_depth>[number]</history_depth> <max_samples>[number]</max_samples> <max_instances>[number]</max_instances> <max_samples_per_instance>[number]</max_samples_per_instance> </durability_service></pre> <p>Example:</p> <pre><durability_service> <service_cleanup_delay> <sec>0</sec> <nanosec>0</nanosec> </service_cleanup_delay> <history_kind>KEEP_LAST_HISTORY_QOS</history_kind> <history_depth>1</history_depth> <max_samples>LENGTH_UNLIMITED</max_samples> <max_instances>LENGTH_UNLIMITED</max_instances> <max_samples_per_instance> LENGTH_UNLIMITED </max_samples_per_instance> </durability_service></pre>
group_data	<p>Attaches arbitrary application data (a buffer of bytes) to discovery meta-data. See documentation on the GroupData QoS policy.</p> <p>Schema:</p> <pre><group_data mode=["hexadecimal" "string"]> <value>[information]</value> </group_data></pre> <p>Example:</p> <pre><group_data mode="string"> <value>Dump information</value> </group_data></pre>

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
latency_budget	<p>Suggests to the middleware how much time is allowed to deliver data. See documentation on the LatencyBudget QoS policy.</p> <p>Schema:</p> <pre><latency_budget> <duration> <sec>[number]</sec> <nanosec>[number]</nanosec> </duration> </latency_budget></pre> <p>Example:</p> <pre><latency_budget> <duration> <sec>0</sec> <nanosec>0</nanosec> </duration> </latency_budget></pre>
lifespan	<p>Specifies how long <i>Connex DDS</i> should consider data sent by an user application to be valid. See documentation on the Lifespan QoS policy.</p> <p>Applies only to <i>DataWriters</i>.</p> <p>Schema:</p> <pre><lifespan> <duration> <sec>[number]</sec> <nanosec>[number]</nanosec> </duration> </lifespan></pre> <p>Example:</p> <pre><lifespan> <duration> <sec>DURATION_INFINITE_SEC</sec> <nanosec> DURATION_INFINITE_NSEC</nanosec> </duration> </lifespan></pre>
liveliness	<p>Liveliness specifies how <i>Connex DDS</i> determines whether a <i>DataWriter</i> is “alive.” See documentation on the Liveliness QoS policy.</p> <p>Schema:</p> <pre><liveliness> <kind> [AUTOMATIC_LIVELINESS_QOS MANUAL_BY_PARTICIPANT_LIVELINESS_QOS MANUAL_BY_TOPIC_LIVELINESS_QOS] </kind> <lease_duration> <sec>[number]</sec> <nanosec>[number]</nanosec> </lease_duration> </liveliness></pre> <p>Example:</p> <pre><liveliness> <kind>AUTOMATIC_LIVELINESS_QOS</kind> <lease_duration> <sec>DURATION_INFINITE_SEC</sec> <nanosec> DURATION_INFINITE_NSEC</nanosec> </lease_duration> </liveliness></pre>

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
multicast	<p>List of multicast addresses on which the <i>DataReader</i> will listen for data. See documentation on the TransportMulticast QoS policy.</p> <p>Applies only to <i>DataReaders</i>.</p> <p>Schema:</p> <pre data-bbox="321 499 954 762"><multicast> <number_of_elements>[number]</number_of_elements> <locator> <kind>[UDPv4]</kind> <address>[IP address]</address> <port>[IP port]</port> </locator> <locator> ... </locator> ... </multicast></pre> <p>Example:</p> <pre data-bbox="321 804 873 1087"><multicast> <number_of_elements>2</number_of_elements> <locator> <kind>UDPv4</kind> <address>192.168.1.0</address> <port>7400</port> </locator> <locator> <kind>UDPv4</kind> <address>192.168.200.0</address> <port>7401</port> </locator> </multicast></pre>
ownership, ownership_ strength	<p>Specifies if <i>DataReaders</i> for a topic can receive data from multiple <i>DataWriters</i> at the same time. See documentation on the Ownership and Ownership Strength QoS policies.</p> <p>Applies to <i>DataReaders</i> (except Strength option) and <i>DataWriters</i></p> <p>Schema:</p> <pre data-bbox="321 1255 987 1423"><ownership> <kind> [SHARED_OWNERSHIP_QOS EXCLUSIVE_OWNERSHIP_QOS] </kind> </ownership> <ownership_strength> <value>[number]</value> </ownership_strength></pre> <p>Example:</p> <pre data-bbox="321 1465 800 1591"><ownership> <kind>EXCLUSIVE_OWNERSHIP_QOS</kind> </ownership> <ownership_strength> <value>0</value> </ownership_strength></pre>

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
partitions	<p>Stores a set of partition names that identify the partitions of which the entity is a member. See documentation on the Partition QoS policy.</p> <p>Schema:</p> <pre data-bbox="321 485 954 705"> <partitions> <number_of_elements>[number]</number_of_elements> <partition> <name>[string]</name> </partition> <partition> ... </partition> ... </partitions> </pre> <p>Example:</p> <pre data-bbox="321 743 873 940"> <partitions> <number_of_elements>2</number_of_elements> <partition> <name>Partition1</name> </partition> <partition> <name>Partition2</name> </partition> </partitions> </pre>
presentation	<p>Controls how <i>Connex DDS</i> presents data received by an application to the <i>DataReaders</i> of the data. See documentation on the Presentation QoS policy.</p> <p>Schema:</p> <pre data-bbox="321 1058 932 1260"> <presentation> <access_scope> [INSTANCE_PRESENTATION_QOS TOPIC_PRESENTATION_QOS GROUP_PRESENTATION_QOS] </access_scope> <coherent_access>[true false]</coherent_access> <ordered_access>[true false]</ordered_access> </presentation> </pre> <p>Example:</p> <pre data-bbox="321 1297 1013 1402"> <presentation> <access_scope>INSTANCE_PRESENTATION_QOS</access_scope> <coherent_access>>false</coherent_access> <ordered_access>>false</ordered_access> </presentation> </pre>

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
property	<p>Stores name/value (string) pairs that can be used to configure certain parameters of <i>Connex DDS</i> that are not exposed through formal QoS policies. See documentation on the Property QoS policy.</p> <p>Schema:</p> <pre> <property> <number_of_elements>[number]</number_of_elements> <element> <name>[string]</name> <value>[string]</value> <propagate>[true false]</propagate> </element> <element> ... </element> ... </property> </pre> <p>Example:</p> <pre> <property> <number_of_elements>2</number_of_elements> <element> <name>Coin</name> <value>Euro</value> <propagate>>false</propagate> </element> <element> <name>Country</name> <value>Spain</value> <propagate>>false</propagate> </element> </property> </pre>
reliability	<p>Determines whether or not data published by a <i>DataWriter</i> will be reliably delivered to matching <i>DataReaders</i>. See documentation on the Reliability QoS policy.</p> <p>Schema:</p> <pre> <reliability> <kind> [RELIABLE_RELIABILITY_QOS BEST_EFFORT_RELIABILITY_QOS] </kind> <max_blocking_time> <sec>[number]</sec> <nanosec>[number]</nanosec> </max_blocking_time> <acknowledgment_kind> [PROTOCOL_ACKNOWLEDGMENT_MODE APPLICATION_AUTO_ACKNOWLEDGMENT_MODE APPLICATION_EXPLICIT_ACKNOWLEDGMENT_MODE] </acknowledgment_kind> </reliability> </pre> <p>Example:</p> <pre> <reliability> <kind> RELIABLE_RELIABILITY_QOS </kind> <max_blocking_time> <sec>0</sec> <nanosec>100000000</nanosec> </max_blocking_time> <acknowledgment_kind> PROTOCOL_ACKNOWLEDGMENT_MODE </acknowledgment_kind> </reliability> </pre>

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
rtps_object_id	<p>Determines the RTPS object ID of a <i>DataWriter/DataReader</i>. See documentation on the DataWriterProtocol QoS policy.</p> <p>Schema: <code><rtps_object_id>[positive number]</rtps_object_id></code></p> <p>Example: <code><rtps_object_id>1</rtps_object_id></code></p>
rtps_protocol_version	<p>Version of the RTPS protocol used by the entity.</p> <p>Schema: <code><rtps_protocol_version></code> <code> [number].[number]</code> <code></rtps_protocol_version></code></p> <p>Example: <code><rtps_protocol_version>0.0</rtps_protocol_version></code></p>
service	<p>Specifies the use of an external persistence service.</p> <p>Schema: <code><service>[NO_SERVICE PERSISTENCE]</service></code></p> <p>Example: <code><service>NO_SERVICE</service></code></p>
time_based_filter	<p>Set by a <i>DataReader</i> to limit the number of new data values received over a period of time. See documentation on the TimeBasedFilter QoS policy.</p> <p>Applies only to <i>DataReaders</i>.</p> <p>Schema: <code><time_based_filter></code> <code> <minimum_separation></code> <code> <sec>[number]</sec></code> <code> <nanosec>[number]</nanosec></code> <code> </minimum_separation></code> <code></time_based_filter></code></p> <p>Example: <code><time_based_filter></code> <code> <minimum_separation></code> <code> <sec>0</sec></code> <code> <nanosec>0</nanosec></code> <code> </minimum_separation></code> <code></time_based_filter></code></p>
topic_data	<p>Attaches arbitrary application data (a buffer of bytes) to discovery meta-data. See documentation on the TopicData QoS policy.</p> <p>Applies only to <i>DataReaders</i>.</p> <p>Schema: <code><topic_data mode=["hexadecimal" "string"]></code> <code> <value>[information]</value></code> <code></topic_data></code></p> <p>Example: <code><topic_data mode="string"></code> <code> <value>Dump information</value></code> <code></topic_data></code></p>

Table 3.1 Configuration Properties for LBED Plugin

Property Name	Property Value and Description
topic_keyed	<p>Indicates whether the data type has a key.</p> <p>Schema: <code><topic_keyed>[true false]</topic_keyed></code></p> <p>Example: <code><topic_keyed>>false</topic_keyed></code></p>
topic_name	<p>The name of the entity's topic. See documentation on the topic parameter in <code>create_datawriter()</code> or <code>create_datareader()</code>.</p> <p>Schema: <code><topic_name>[string]</topic_name></code></p> <p>Example: <code><topic_name>Example topic</topic_name></code></p>
type_name	<p>The name of the entity's type. See documentation on the type_name parameter in <code>create_topic()</code>.</p> <p>Schema: <code><type_name>[string]</type_name></code></p> <p>Example: <code><type_name>Example type</type_name></code></p>
unicast	<p>List of all unicast addresses on which the <i>DataReader/DataWriter</i> will listen for data. See documentation on the TransportUnicast QoS policy.</p> <p>Schema: <code><unicast></code> <code> <number_of_elements>[number]</number_of_elements></code> <code> <locator></code> <code> <kind>[UDIPv4]</kind></code> <code> <address>[IP address]</address></code> <code> <port>[IP port]</port></code> <code> </locator></code> <code> <locator></code> <code> ...</code> <code> </locator></code> <code> ...</code> <code></unicast></code></p> <p>Example: <code><unicast></code> <code> <number_of_elements>2</number_of_elements></code> <code> <locator></code> <code> <kind>UDIPv4</kind></code> <code> <address>82.123.32.4</address></code> <code> <port>7400</port></code> <code> </locator></code> <code> <locator></code> <code> <kind>UDIPv4</kind></code> <code> <address>82.123.32.7</address></code> <code> <port>7401</port></code> <code> </locator></code> <code></unicast></code></p>
user_data	<p>Attaches arbitrary application data (a buffer of bytes) to discovery meta-data. See documentation on the UserData QoS policy.</p> <p>Schema: <code><user_data mode=["hexadecimal" "string"]></code> <code> <value>[information]</value></code> <code></user_data></code></p> <p>Example: <code><user_data mode="hexadecimal"></code> <code> <value>04FC8D922E</value></code> <code></user_data></code></p>

3.2 Configuring LBED Plugin in Connex DDS

This section describes how to configure the properties for the LBED plugin in the XML QoS Profile file used by *Connex DDS* (such as **USER_QOS_PROFILES.XML**), or in the PropertyQosPolicy for your application's *DomainParticipant*.

Let's look at part of an example XML file, which you can find in **<path to examples>/connex_dds/c/limited_bandwidth_plugins/lbediscovery/USER_QOS_PROFILES.xml**:

```
<qos_profile name="LBEDiscoveryPluginExampleSubscriber_Profile">
  ...
  <participant_qos>
    <property>
      <value>
        <element>
          <name>
            dds.discovery.endpoint.lbediscovery.library
          </name>
          <value>rtilbedisc</value>
        </element>
        <element>
          <name>
            dds.discovery.endpoint.lbediscovery.create_function
          </name>
          <value>DDS_LBEDiscoveryPlugin_create</value>
        </element>
        <element>
          <name>
            dds.discovery.endpoint.lbediscovery.config_file
          </name>
          <value>
            LBEDiscoveryPluginExamplePublisher.xml
          </value>
        </element>
        <element>
          <name>dds.discovery.endpoint.load_plugins</name>
          <value>dds.discovery.endpoint.lbediscovery</value>
        </element>
      </value>
    </property>
  </participant_qos>
  ...

```

[Table 3.2 LBED Configuration Properties for Connex DDS](#) describes the name/value pairs that you can use to configure the LBED plugin.

Table 3.2 LBED Configuration Properties for Connex DDS

Property Name	Property Value and Description
dds.discovery.endpoint.load_plugins	<p>Required.</p> <p>String indicating the prefix name of the plugin that will be loaded by <i>Connex DDS</i>.</p> <p>Set the value to dds.discovery.endpoint.<string>, where <i><string></i> can be any string you want, as long as you use the same string consistently for all the properties in this table. Our example uses <code>lbediscovery</code>:</p> <pre><element> <name>dds.discovery.endpoint.load_plugins</name> <value>dds.discovery.endpoint.lbediscovery</value> </element></pre>
dds.discovery.endpoint.<string>.library	<p>Required.</p> <p>The name of the dynamic library that contains the LBED plugin implementation. This library must be in the path during run time for use by <i>Connex DDS</i>.</p> <p>Set the value to rtilbedisc.</p> <p>Example:</p> <pre><element> <name>dds.discovery.endpoint.lbediscovery.library</name> <value>rtilbedisc</value> </element></pre>
dds.discovery.endpoint.<string>.create_function	<p>Required.</p> <p>The name of the function that will be called by <i>Connex DDS</i> to create an instance of the LBED plugin.</p> <p>Set the value to DDS_LBEDiscoveryPlugin_create.</p> <p>Example:</p> <pre><element> <name> dds.discovery.endpoint.lbediscovery.create_function </name> <value>DDS_LBEDiscoveryPlugin_create</value> </element></pre>
dds.discovery.endpoint.<string>.config_file	<p>Required.</p> <p>The name of the discovery configuration file, described in 3.1 Creating the LBED Plugin Configuration File on page 5.</p> <p>Set the value to the name of your own file.</p> <p>Example:</p> <pre><element> <name> dds.discovery.endpoint.lbediscovery.config_file </name> <value>LBEDiscoveryPluginExampleSubscriber.xml</value> </element></pre>

Table 3.2 LBED Configuration Properties for Connex DDS

Property Name	Property Value and Description
dds.discovery.endpoint. <string>.verbosity	<p>The verbosity for the plugin, for debugging purposes.</p> <ul style="list-style-type: none"> • -1: Silent • 0: Exceptions only (default) • 1: Warnings • 2 and up: Debug <p>Example:</p> <pre> <element> <name> dds.discovery.endpoint.lbediscovery.verbosity </name> <value>0</value> </element> </pre>

3.3 Optimizing the Plugin

You can reduce network bandwidth by changing some *Connex DDS* properties in the XML file, **USER_QOS_PROFILE.xml**. You can find an example of this user profile in <path to examples>/connex_dds/c/limited_bandwidth_plugins/lbediscovery/USER_QOS_PROFILES.xml.