

LLRP HelloWorld Quick Start Guide

CHANGE LOG

7/25/07 -- Document created to support 1.4 beta release
8/05/07 -- Documented Updated to reflect 1.4 release
10/19/07 Document Updated to reflect 1.4.2 release
1/25/08 Document Updated to reflect 1.5 release

Table of Contents

PURPOSE.....	1
SHORT INTRODUCTION TO LLRP.....	1
LLRP CONNECTION.....	2
LLRP VIRTUAL READER SUPPORTED COMMANDS.....	2
PREREQUISITES.....	2
SETUP OF LLRP HELLO WORLD CLIENT.....	3
EXECUTION.....	3
Stage 1 - Client Connects.....	3
Stage 2 - Reader Capabilities.....	3
Stage 3 - Reader Configuration.....	3
Stage 4 Reading Tags.....	4
Stage 5 Writing To tags.....	4
Stage 6 Clean up.....	5
RESULTS.....	5
Stage 1.....	5
Stage 2.....	5
Stage 3.....	7
Stage 4.....	8
Stage 5.....	11
Stage 6.....	13
TERMINOLOGY.....	14
USEFUL RESOURCES.....	14
MAILING LISTS.....	14

PURPOSE

The purpose of the guide is to provide a HelloWorld example demonstrating the core functions associated with the LLRP Specification.

SHORT INTRODUCTION TO LLRP

There has been a lot of buzz in the RFID world as of late around LLRP (Low Level Reader Protocol). It is an EPC Ratified Standard that describes a protocol for client to reader communication. It is called "Low Level" because it allows control of Air Protocol (such as class 1 gen 2) parameters as well as control of other hardware aspects of the reader. LLRP communicates via Messages -- the basic data unit -- that are encoded in a binary format and are sent over TCP. Using these messages, the client can control the various dynamic data structures (ROSpecs, AccessSpecs, etc.) and configure the reader. Because these messages are encoded in a binary format, there is no way to telnet to the reader and issue commands. This means that it is required to have a client (such as this HelloWorld client) to form and issue commands.

There are two main data structures defined in the LLRP spec. The first is the ROSpec (Reader Operations Specification). The most important aspect of the ROSpec is that it contains a list of one or more AISpecs (which define how to read EPC tags from the antennas) and RFSurveySpec (which define how to get operation information about the antennas such as power levels). In addition the ROSpec contains a ReportSpec that has the parameters describing what information a report will contain. The other main data structure is the AccessSpec which defines how to read non-EPC information on a tag (such as user defined data) as well as how to perform other operations on a tag, such as writing and killing. The AccessSpec contains a tag pattern that it uses to determine which tags it should operate on. Additionally, it has a list of OpSpecs that define the operations (e.g. Read, write, lock kill) that it should perform on the tags.

It may help to imagine ROSpecs as processes inside an operating system. A ROSpec is like a single process, and it moves between three states -- disabled, inactive, and active. When the ROSpec is added to the reader, it is added in the disabled state. Then, a message from the client --the ENABLE_ROSPEC message -- moves the ROSpec to the inactive state. After that, a trigger event (time based, message based, etc) moves the ROSpec into the active state, where it begins executing. When the ROSpec is executing, it goes through its list of specs to executes (AI and/or RFSurvey) and executes each one sequentially. It keeps repeating this cycle until its ROSpec stop trigger fires, at which time it moves back to the inactive state. The AccessSpecs execute concurrently with the ROSpec. After an AISpec has identified the tags that are in the reader's field of view, the access spec will determine if one or more of the tags match its tag pattern. If they do, it will execute the OpSpecs that are defined in it on the tags.

For more information, please see the EPC LLRP Spec at <http://www.epcglobalinc.org/standards/llrp>

LLRP CONNECTION

The LLRP PReader has the ability to either open up a port for incoming connections (server mode) or to establish a connection with the client (client mode). Therefore when the rifidi LLRP reader is started, an admin console is started at the administration IP. To access this interface, simply telnet into it, and type help for information. However, if you would like the reader to start up in server mode automatically, simply check the box above and specify an LLRP Connection Address.

LLRP VIRTUAL READER SUPPORTED COMMANDS

- 1) ADD_ROSPEC message
- 2) ENABLE_ROSPEC message
- 3) START_ROSPEC message

- 4) STOP_ROSPEC message
- 5) DISABLE_ROSPEC message
- 6) DELETE ROSPEC message
- 7) GET_ROSPECS message
- 8) GET_READER_CAPABILITIES message
- 9) GET_READER_CONFIG message
- 10) SET_READER_CONFIG message
- 11) ADD_ACCESSSPEC message
- 12) ENABLE_ACCESSSPEC message
- 13) DISABLE_ACCESSSPEC message
- 14) DELETE_ACCESSSPEC message
- 15) GET_ACCESSSPECS message
- 16) CLOSE_CONNECTION message
- 17) GET_REPORT message

PREREQUISITES

- Installation of Rifidi 1.5
- LLRPHelloWorld Client - see the LLRPHelloWorld directory.
- java 1.5 and javac installed.

SETUP OF LLRP HELLO WORLD CLIENT

- 1) Configure LLRP Virtual Reader in Rifidi 1.5 IDE
 - a) Start Rifidi
 - b) Click the 'create a new reader' button
 - c) Choose 'LLRP Reader'
 - d) Enter a name. Make sure the check box is checked next to 'enable server mode at startup'
 - e) Check the box to enable GPIO & click next
 - f) Click Finish
 - g) Add a Gen 2 tag to the reader by creating it, and dragging it onto the reader's field of view
 - h) Right click on the reader and click start
- 2) Configure the LLRP Hello World Client
 - a) Open up a terminal and 'cd' to the LLRPHelloWorld directory
 - b) type 'javac -cp llrp.jar:. LLRPHelloWorldClient.java' if you are in linux or
'javac -cp llrp.jar;. LLRPHelloWorldClient.java' if you are in windows to compile the code
Note that this step will also automatically compile ByteAndHexConvertingUtility.java, a Utility class used for printing out Hex values properly.
 - c) type 'java -cp llrp.jar:. LLRPHelloWorldClient ' if you are on linux or
'java -cp llrp.jar;. LLRPHelloWorldClient ' if you are on windows to run the code.

Note that the default IP address and port number for the Rifidi LLRP Reader is 127.0.0.1 and 5084, respectively. If you are using a different IP or port, you will need to specify them in step 2.c. Simply add them as command line arguments, like this:

```
'java -cp llrp.jar;. LLRPHelloWorldClient 127.0.0.1 5084 '
```

The results will be written out to a text file in the directory from which you executed the client.

EXECUTION

Stage 1 - Client Connects

In this stage, the client connects to the reader at the specified IP and port. The reader will send a connection event notification message. Messages may now be sent between the client and the reader.

Stage 2 - Reader Capabilities

- 1) GET_READER_CAPABILITIES Message

This will cause the reader to give back its capabilities. The reader's capabilities are variables that tell what the reader can and cannot do. They may not be set by the client.

Stage 3 - Reader Configuration

- 1) SET_READER_CONFIG Message

In this step, the client will send the reader a SET_READER_CONFIG message that will set some global variables such as when to send reader event notifications, and what information the reports should contain.

2) GET_READER_CONFIG Message

This will cause the reader to respond back with its configuration variables. These variables may be set by the client with a get reader configuration. Although the reader is able to get and set these variables at the moment (with the exception of those mentioned in the release notes), they do not actually affect the rfid reader. They will be more fully integrated in a later release)

Stage 4 Reading Tags

1) ADD_ROSPEC Message

In this step, the client constructs the ROSpec object, which consists of numerous parameters specifying how the reader should read tags and send back results to the client. It is only through a ROSpec that a client can set up a reader. There is no concept of "telnetting" to the reader and using a command line in LLRP. The newly created ROSpec, is sent to the reader which adds it to its list of "ROSpecs to be processed." When a ROSpec is added, it is added to the reader's "disabled" queue.

2) GET_ROSPEC Message

It is possible to see all of the ROSpecs that are defined in the reader using the GET_ROSPEC Message.

3) ENABLE_ROSPEC Message

A ROSpec must be enabled before it can be executed. The ENABLE_ROSPEC is a message sent from the client to the reader that moves the ROSpec from the "disabled" queue to the reader's "enabled" queue

4) START_ROSPEC Message

Once a ROSpec is enabled, it is listening for a start trigger. The simplest way to issue a start trigger is to send the START_ROSPEC message which will cause a reader to begin execution.

5) Receive Reports from reader

When a ROSpec is in the execution phase, it can begin to send reports back to the client. Reports contain the information about the tags that the reader has seen, such as the EPC number, the first and last seen time, and the number of times the tag has been seen. The reader will also send back ROSpec and AISpec Event notifications, if those were enabled through the SET_READER_CONFIG message.

6) ROSpec Stops

A ROSpec stop event triggers the end of the execution of the ROSpec. In this case, the rospec is set to listen for a duration trigger that is the rospec will execute for a predefined amount of time. However, another stop trigger that can be used is the ROSPEC_STOP Message sent from the client to the reader. Once the ROSpec stops, it moves back into the inactive state, and will wait for another start trigger.

Stage 5 Writing To tags

1) ADD_ACCESSSPEC Message

The access spec is the specification of how a reader should perform advanced operations (such as reading tag memory, or writing to tag memory) on a tag. See the overview of llrp section in this document for an explanation of how access specs work.

2) GET_ACCESSSPECS Message

It is possible to see all the AccessSpecs that are defined in the reader by using this message.

3) ENABLE_ACCESSSPEC Message

Before an access spec can perform any operations on the tag, it must be enabled first.

4) START_ROSPEC Message

The access spec cannot run on its own; it requires a ROSpec to be executing. In this step we restart the rospec that is already defined in the reader. Then, if an AISpec in the ROSpec finds tags in the reader's field of view that match the Access spec's tag pattern, the access spec will execute its list of opspects on them.

5) Receive Reports from reader

Now the Reports from the reader have an OpSpecResult in them that gives the results of the access specs that have run. In this case, there was only one opspect in the access spec, and it was set up to write a new EPC ID to any tag that the reader saw. However, it was also set up to only execute once. So the report may send back two tags: one tag that it saw originally, on which the opspect performed, and then a new tag that has the ID that the access spec wrote.

6) ROSpec Stops

Stage 6 Clean up

1) DISABLE_ROSPEC Message

This will cause a ROSpec to be disabled. Once the ROSpec is disabled, it must be enabled again before it can listen for start triggers

2) DELETE_ROSPEC Message

This will cause a RoSpec to be deleted. Once the ROSpec is deleted, it cannot be enabled or started again.

RESULTS

Stage 1

```
<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="1192819103578000"/>
    </UTCTimestamp>
    <ConnectionAttemptEvent>
      <Status value="0"/>
    </ConnectionAttemptEvent>
  </ReaderEventNotificationData>
</Message>
```

Stage 2

```
<Message type="GetReaderCapabilitiesResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="Success"/>
  </LLRPStatus>
  <GeneralDeviceCapabilities>
    <MaxNumberOfAntennaSupported value="1"/>
    <CanSetAntennaProperties value="false"/>
    <HasUTCClockCapability value="false"/>
    <DeviceManufacturerName value="0"/>
    <ModelName value="0"/>
    <ReaderFirmwareVersion value="LLRP_Reader1.4"/>
    <ReceiveSensitivityTableEntry>
      <Index value="0"/>
      <ReceiveSensitivityValue value="0"/>
    </ReceiveSensitivityTableEntry>
    <GPIOCapabilities>
      <NumGPIOs value="1"/>
      <NumGPOs value="1"/>
    </GPIOCapabilities>
    <PerAntennaAirProtocol>
      <AntennaID value="1"/>
      <ProtocolElement value="1"/>
    </PerAntennaAirProtocol>
  </GeneralDeviceCapabilities>
  <LLRPCapabilities>
    <CanDoRFSurvey value="false"/>
    <CanReportBufferFillWarning value="false"/>
    <SupportsClientRequestOpSpec value="false"/>
    <CanDoTagInventoryStateAwareSingulation value="false"/>
    <SupportsEventAndReportHolding value="false"/>
    <MaxPriorityLevelSupported value="0"/>
    <ClientRequestOpSpecTimeout value="0"/>
    <MaxNumROSpecs value="1"/>
  </LLRPCapabilities>
</Message>
```

```

<MaxNumSpecsPerROSpec value="1"/>
<MaxNumInventoryParameterSpecsPerAISpec value="0"/>
<MaxNumAccessSpecs value="0"/>
<MaxNumOpSpecsPerAccessSpec value="0"/>
</LLRPCapabilities>
<RegulatoryCapabilities>
  <CountryCode value="0"/>
  <CommunicationsStandard value="0"/>
  <UHFBandCapabilities>
    <TransmitPowerLevelTableEntry>
      <Index value="0"/>
      <TransmitPowerValue value="0"/>
    </TransmitPowerLevelTableEntry>
    <FrequencyInformation>
      <Hopping value="false"/>
      <FixedFrequencyTable>
        <FrequencyElement value="0"/>
      </FixedFrequencyTable>
    </FrequencyInformation>
  <UHFC1G2RFModeTableEntry>
    <ModeID value="0"/>
    <DRValue value="false"/>
    <Conformance value="false"/>
    <Mod value="0"/>
    <FLM value="0"/>
    <M value="0"/>
    <BDRValue value="0"/>
    <PIEValue value="0"/>
    <MinTariValue value="0"/>
    <MaxTariValue value="0"/>
    <StepTariValue value="0"/>
  </UHFC1G2RFModeTableEntry>
</UHFBandCapabilities>
</RegulatoryCapabilities>
<C1G2LLRPCapabilities>
  <CanSupportBlockErase value="false"/>
  <CanSupportBlockWrite value="false"/>
  <MaxNumSelectFiltersPerQuery value="0"/>
</C1G2LLRPCapabilities>
</Message>

```

Stage 3

```

<Message type="SetReaderConfigResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value=""/>
  </LLRPStatus>

```


</Message>

<Message type="GetReaderConfigResponse">

<LLRPStatus>

<ErrorCode value="0"/>

<ErrorDescription value="Success"/>

</LLRPStatus>

<Identification>

<IDType value="0"/>

<ReaderIDElement value="0"/>

<ReaderIDElement value="0"/>

<ReaderIDElement value="0"/>

<ReaderIDElement value="0"/>

<ReaderIDElement value="0"/>

<ReaderIDElement value="0"/>

</Identification>

<AntennaProperties>

<AntennaConnected value="false"/>

<AntennaId value="1"/>

<AntennaGain value="0"/>

</AntennaProperties>

<AntennaConfiguration>

<AntennaId value="1"/>

<RFReceiver>

<ReceiverSensitivity value="0"/>

</RFReceiver>

<RFTransmitter>

<HopTableId value="0"/>

<ChannelIndex value="0"/>

<TransmitPower value="0"/>

</RFTransmitter>

</AntennaConfiguration>

<ReaderEventNotificationSpec>

<EventNotificationState>

<EventType value="0"/>

<NotificationState value="false"/>

</EventNotificationState>

<EventNotificationState>

<EventType value="1"/>

<NotificationState value="false"/>

</EventNotificationState>

<EventNotificationState>

<EventType value="2"/>

<NotificationState value="true"/>

</EventNotificationState>

<EventNotificationState>

<EventType value="3"/>

<NotificationState value="false"/>

</EventNotificationState>

```
<EventNotificationState>
  <EventType value="4"/>
  <NotificationState value="false"/>
</EventNotificationState>
<EventNotificationState>
  <EventType value="5"/>
  <NotificationState value="false"/>
</EventNotificationState>
<EventNotificationState>
  <EventType value="6"/>
  <NotificationState value="true"/>
</EventNotificationState>
<EventNotificationState>
  <EventType value="7"/>
  <NotificationState value="false"/>
</EventNotificationState>
<EventNotificationState>
  <EventType value="8"/>
  <NotificationState value="false"/>
</EventNotificationState>
</ReaderEventNotificationSpec>
<ROReportSpec>
  <ROReportTrigger value="2"/>
  <N value="0"/>
  <TagReportContentSelector>
    <EnableROSpecID value="true"/>
    <EnableSpecIndex value="true"/>
    <EnableInventoryParameterSpecID value="true"/>
    <EnableAntennaID value="true"/>
    <EnableChannelIndex value="true"/>
    <EnablePeakRSSI value="true"/>
    <EnableFirstSeenTimestamp value="true"/>
    <EnableLastSeenTimestamp value="true"/>
    <EnableTagSeenCount value="true"/>
    <EnableAccessSpecID value="true"/>
    <C1G2EPCMemorySelector>
      <EnableCRC value="false"/>
      <EnablePCBits value="false"/>
    </C1G2EPCMemorySelector>
  </TagReportContentSelector>
</ROReportSpec>
<AccessReportSpec>
  <AccessReportTrigger value="0"/>
</AccessReportSpec>
<LLRPConfigurationStateValue>
  <LLRPConfigurationStateValue value="17"/>
</LLRPConfigurationStateValue>
<KeepaliveSpec>
  <KeepaliveTriggerType value="0"/>
```

```
<TimeInterval value="0"/>
</KeepaliveSpec>
<EventsAndReports>
  <HoldEventsAndReportsUponReconnect value="false"/>
</EventsAndReports>
</Message>
```

Stage 4

```
<Message type="AddROSpecResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="ROSpec successfully added"/>
  </LLRPStatus>
</Message>
```

```
<Message type="GetROSpecsResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="sucess"/>
  </LLRPStatus>
  <ROSpec>
    <ROSpecID value="1"/>
    <Priority value="0"/>
    <CurrentState value="0"/>
    <ROBoundarySpec>
      <ROSpecStartTrigger>
        <ROSpecStartTriggerType value="0"/>
      </ROSpecStartTrigger>
      <ROSpecStopTrigger>
        <ROSpecStopTriggerType value="1"/>
        <DurationTriggerValue value="5000"/>
      </ROSpecStopTrigger>
    </ROBoundarySpec>
    <AISpec>
      <AntennaElement value="1"/>
      <AISpecStopTrigger>
        <AISpecStopTriggerType value="0"/>
        <DurationTrigger value="0"/>
      </AISpecStopTrigger>
      <InventoryParameterSpec>
        <InventoryParameterSpecID value="1"/>
        <ProtocolID value="1"/>
      </InventoryParameterSpec>
    </AISpec>
  </ROSpec>
</Message>
```

```
<Message type="EnableROSpecResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="ROSpec was successfully enabled"/>
  </LLRPStatus>
</Message>
```

```
<Message type="StartROSpecResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="ROSpec was successfully started"/>
  </LLRPStatus>
</Message>
```

```
<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="1192819103812000"/>
    </UTCTimestamp>
    <ROSpecEvent>
      <EventType value="0"/>
      <ROSpecID value="1"/>
      <PreemptingROSpecID value="0"/>
    </ROSpecEvent>
  </ReaderEventNotificationData>
</Message>
```

```
<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="1192819109062000"/>
    </UTCTimestamp>
    <AISpecEvent>
      <EventType value="0"/>
      <ROSpecID value="1"/>
      <SpecIndex value="0"/>
    </AISpecEvent>
  </ReaderEventNotificationData>
</Message>
```

```
<Message type="ROAccessReport">
  <TagReportData>
    <EPCCData>
      <data value="#####"/>
    </EPCCData>
    <ROSpecID>
      <ROSpecID value="1"/>
    </ROSpecID>
    <SpecIndex>
```

```

    <SpecIndex value="0"/>
  </SpecIndex>
  <InventoryParameterSpecID>
    <InventoryParamSpecID value="0"/>
  </InventoryParameterSpecID>
  <AntennaID>
    <AntennaID value="1"/>
  </AntennaID>
  <PeakRSSI>
    <PeakRSSI value="0"/>
  </PeakRSSI>
  <ChannelIndex>
    <ChannelIndex value="0"/>
  </ChannelIndex>
  <FirstSeenTimestampUTC>
    <Microseconds value="1192817830781000"/>
  </FirstSeenTimestampUTC>
  <LastSeenTimestampUTC>
    <Microseconds value="1192819103812000"/>
  </LastSeenTimestampUTC>
  <TagSeenCount>
    <TagCount value="1"/>
  </TagSeenCount>
</TagReportData>
</Message>

```

```

<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="1192819109062000"/>
    </UTCTimestamp>
    <ROSpecEvent>
      <EventType value="1"/>
      <ROSpecID value="1"/>
      <PreemptingROSpecID value="0"/>
    </ROSpecEvent>
  </ReaderEventNotificationData>
</Message>

```

Stage 5

```

<Message type="AddAccessSpecResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="Success"/>
  </LLRPStatus>
</Message>

```

```
<Message type="GetAccessSpecsResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="Success"/>
  </LLRPStatus>
  <AccessSpec>
    <AccessSpecID value="2"/>
    <AntennaId value="0"/>
    <ProtocolId value="1"/>
    <CurrentState value="false"/>
    <ROSpecID value="0"/>
    <AccessSpecStopTrigger>
      <AccessSpecStopTrigger value="1"/>
      <OperationCountValue value="1"/>
    </AccessSpecStopTrigger>
    <AccessCommand>
      <C1G2TagSpec>
        <C1G2TargetTag>
          <MB value="2"/>
          <Match value="true"/>
          <Pointer value="3"/>
          <Mask value=""/>
          <Data value=""/>
        </C1G2TargetTag>
      </C1G2TagSpec>
      <C1G2Write>
        <OpSpecID value="2"/>
        <AccessPassword value="0"/>
        <MB value="1"/>
        <WordPointer value="2"/>
        <WriteDataElement value="1"/>
        <WriteDataElement value="2"/>
        <WriteDataElement value="3"/>
        <WriteDataElement value="4"/>
        <WriteDataElement value="5"/>
        <WriteDataElement value="6"/>
        <WriteDataElement value="7"/>
        <WriteDataElement value="8"/>
        <WriteDataElement value="9"/>
        <WriteDataElement value="10"/>
        <WriteDataElement value="11"/>
        <WriteDataElement value="12"/>
      </C1G2Write>
    </AccessCommand>
  </AccessSpec>
</Message>

<Message type="EnableAccessSpecResponse">
  <LLRPStatus>
```

```
<ErrorCode value="0"/>
<ErrorDescription value="Success"/>
</LLRPStatus>
</Message>

<Message type="StartROSpecResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="ROSpec was successfully started"/>
  </LLRPStatus>
</Message>

<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="1192819109750000"/>
    </UTCTimestamp>
    <ROSpecEvent>
      <EventType value="0"/>
      <ROSpecID value="1"/>
      <PreemptingROSpecID value="0"/>
    </ROSpecEvent>
  </ReaderEventNotificationData>
</Message>

<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="1192819115000000"/>
    </UTCTimestamp>
    <AISpecEvent>
      <EventType value="0"/>
      <ROSpecID value="1"/>
      <SpecIndex value="0"/>
    </AISpecEvent>
  </ReaderEventNotificationData>
</Message>

<Message type="ROAccessReport">
  <TagReportData>
    <EPCData>
      <data value="#####"/>
    </EPCData>
    <ROSpecID>
      <ROSpecID value="1"/>
    </ROSpecID>
    <SpecIndex>
      <SpecIndex value="0"/>
    </SpecIndex>
```

```

<InventoryParameterSpecID>
  <InventoryParamSpecID value="0"/>
</InventoryParameterSpecID>
<AntennaID>
  <AntennaID value="1"/>
</AntennaID>
<PeakRSSI>
  <PeakRSSI value="0"/>
</PeakRSSI>
<ChannelIndex>
  <ChannelIndex value="0"/>
</ChannelIndex>
<FirstSeenTimestampUTC>
  <Microseconds value="1192817830781000"/>
</FirstSeenTimestampUTC>
<LastSeenTimestampUTC>
  <Microseconds value="1192819109765000"/>
</LastSeenTimestampUTC>
<TagSeenCount>
  <TagCount value="1"/>
</TagSeenCount>
<C1G2WriteOpSpecResult>
  <Result value="0"/>
  <OpSpecID value="2"/>
  <NumWordsWritten value="6"/>
</C1G2WriteOpSpecResult>
</TagReportData>
</Message>

```

```

<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="1192819115000000"/>
    </UTCTimestamp>
    <ROSpecEvent>
      <EventType value="1"/>
      <ROSpecID value="1"/>
      <PreemptingROSpecID value="0"/>
    </ROSpecEvent>
  </ReaderEventNotificationData>
</Message>

```

Stage 6

```

<Message type="DisableROSpecResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="ROSpec was sucessfully disabled"/>
  </LLRPStatus>
</Message>

```



```

</LLRPStatus>
</Message>

<Message type="DeleteROSpecResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value="ROSpec was successfully deleted"/>
  </LLRPStatus>
</Message>

<Message type="CloseConnectionResponse">
  <LLRPStatus>
    <ErrorCode value="0"/>
    <ErrorDescription value=""/>
  </LLRPStatus>
</Message>

<Message type="ReaderEventNotification">
  <ReaderEventNotificationData>
    <UTCTimestamp>
      <Microseconds value="0"/>
    </UTCTimestamp>
    <ConnectionCloseEvent>
      </ConnectionCloseEvent>
    </ReaderEventNotificationData>
  </Message>

```

TERMINOLOGY

AISPEC (Antenna Inventory Specification) - An AISpecs is the basic unit of execution. It contains parameters specifying how a tag should be read from the antennas.

ROSPEC (Reader Operation Specification) - A ROSpec is a container for AISpecs. It also contains parameters that specify how long the ROSpec should run and what information should be sent back in the reports. See the state diagram on page 23 of the EPC LLRP spec for information about how a rospec is executed.

OPSPEC (Operation Specification) - An opSpec defines how a single advanced operation (such as write, lock, and kill) should be performed on a tag. For example, for a write operation, it defines the memory bank to use, word pointer to start writing at, and the bytes to write to the tag.

ACCESSSPEC (Access Specification) - An accessSpec contains a list of OpSpecs. Additionally, it contains a tag mask that is used to determine on which tags the opsspecs should be executed.

MESSAGE - A message is a unit of communication between the reader and the client. All messages are sent over TCP and are encoded in binary. The use of messages are the only way to communicate to the reader, so there is no way to telnet to a reader and issue commands.

REPORT - A report is a message that is sent back from the reader that contains a list of tags that were read.

CLIENT - A client is a program that connects to the LLRP reader to control it via messages

USEFUL RESOURCES

<http://www.rifidi.org>

<http://www.llrp.org>

<http://www.epcglobalinc.org>

<http://sourceforge.net/projects/llrp-toolkit/>

<http://sourceforge.net/projects/rifidi/>

MAILING LISTS

rifidi-users@lists.sourceforge.net

rifidi-developers@lists.sourceforge.net

llrp-toolkit-devel@lists.sourceforge.net