1. How is LLRP different from RP v1.1?

LLRP is the acronym for Low Level Reader Protocol. RP is the acronym for Reader Protocol. The biggest distinction between the two is that LLRP is "RFID air protocol aware" while RP v1.1 provides a higher level abstraction that hides many RFID details. In fact, RP v1.1 is specified to accommodate non-RFID data collection devices such as bar code.

LLRP is called low-level because it provides control of RFID air protocol operation timing and access to air protocol command parameters. The design of this interface recognizes that in some RFID systems, there is a requirement for explicit knowledge of RFID air protocols and the ability to control Readers that implement RFID air protocol communications. It also recognizes that coupling control to the physical layers of an RFID infrastructure may be useful for the purpose of mitigating RFID interference.

Unlike other automated data collection systems such as bar code, an RFID system is essentially a wireless infrastructure. LLRP was designed to provide direct control over the wireless aspects of an RFID reader. RP v1.1 was designed to provide a higher level interface that abstracts these details from application software. Therefore, RP v1.1 is not appropriate for software required to specifically control the wireless aspects of an RFID reader.

It should be noted that like RP v1.1, ALE also provides a high level interface for EPC data.

2. What are the end-points of an LLRP connection?

LLRP has two end-points: the Reader end-point and the Client end-point.

The LLRP workgroup deliberated the naming of the Client end-point. In the first draft of the LLRP spec, the Client end-point was, instead, called the "Host" end-point. There was an objection that the term "Host" would mislead people into thinking of a host IT system such as an ERP system. So, it was recommended that we call it the "Controller" end-point. But, other people were concerned that the word "Controller" would imply that this end-point had to be a hardware device.

However, the intent is that it can be either hardware or software. As a compromise, now in the LLRP spec it's called the "Client" end-point. So, anything that talks to a LLRP Reader is simply called a "Client".

3. **What tools are available or planned to help me implement LLRP?**

   The primary normative reference to implement LLRP is of course, the ratified EPCglobal specification, this FAQ, and for EPCglobal subscribers, the LLRP conformance document. However, several organizations are providing open source efforts using LLRP including:

   http://www.llrp.org (A programming toolkit for building and parsing LLRP messages)
   http://www.Rifidi.org (Rifidi open source IDE for RFID)
   http://www.Accada.org (Open source EPCglobal prototyping platform)
   http://tag-centric.sourceforge.net/ (Open source RFID edgeware)
   http://wireshark.org (Open source LAN analyzer including a plug-in for LLRP)

4. **Using LLRP, how do I write an EPC on a tag?**

   LLRP, being a low level interface, provides access to the raw binary data on tags. Therefore, to write an EPC you'll first need the EPC represented in terms of raw data. See the Tag Data Specification v1.3 (http://www.epcglobalinc.org/standards/tds/tds_1_3-standard-20060308.pdf) section 4.3.9.

   The C1G2Write parameter (an air protocol OpSpec parameter) is used to write data to a Class-1 Gen-2 (C1G2) tag. See section 15.2.1.3.2.2 of the LLRP spec (http://www.epcglobalinc.org/standards/llrp/llrp_1_0_1-standard-20070813.pdf).

   The EPC resides in memory bank 1 beginning at word address 2. The tag's EPC memory bank may be write protected. If it's permanently locked, then you won't be able to write it. If it's password protected, then you'll need to provide the correct password when writing this memory bank.

   When writing the EPC memory bank you should also be familiar with the tag's "Protocol Control" (PC) bits. See the (C1G2) spec (http://www.epcglobalinc.org/standards/uhfc1g2/uhfc1g2_1_0_9-standard-20050126.pdf) section 6.3.2.1.4.

The 5-most MSBs of the PC define the number of words (16-bits) that the tag will backscatter when it is singulated. The remaining bits are RFU (reserved) and NSI. Currently, both RFU and NSI bits are always zero for EPC-encoded tags (i.e., tags encoded according to the EPCglobal TDS).

As an example, a 96-bit EPC equates to 6 words (16 bits). So, when writing any 96-bit EPC the PC should be written as 0x3000 (i.e., length=6, NSI=0).

5. **Can LLRP be used to access non-EPC data?**

   LLRP v1.0 supports the EPCglobal UHF Class-1 Generation-2 (C1G2) air protocol. The C1G2 standard was adopted by ISO as ISO 18000-6C and therefore LLRP can be used to access ISO 18000-6C tags.

   Refer to the C1G2 spec (see URL provided with the answer to question #4) section 6.3.2.1.4 which defines the numbering system identifier (NSI) of a tag. If the MSB of the NSI (bit 17_h of the EPC memory bank) contains a logical 1, then the tag conforms to ISO 18000-6C and its references to ISO tag data encoding standards.

   Note: if you're writing tags for a closed-system environment, there's a simple way to insure that your tags are properly recognized as closed-system tags even should your tags find their way into an open-system environment. ISO 15693-3 assigns AFI values 1-3 for closed-system environments. Therefore, when writing a proprietary encoding on a closed-system tag, if you write the tag's NSI=0x101, 0x102, or 0x103 then the RFID community will know that your tag is intended for a closed-system. The tag should be ignored by open-system RFID applications.

6. **I hear that LLRP is air-protocol aware. What air protocol parameters do I *have* to configure to use LLRP? Can I get tag data without specifying air-protocol parameters?**

   LLRP exposes air-protocol parameters such as gen2 mode, Transmit power, etc. However, LLRP has only one required air-protocol parameter in its inventory configuration -- the identifier of the air-protocol to use for operations. LLRP 1.0.1 supports only the UHF C1G2 air protocol.

7. What is a "spec" and what does this mean in LLRP?

LLRP Client implementations pass sets of "rules" to the Reader implementation via the LLRP protocol. These "rules" or "specs" as they are called in LLRP allow the Client to pre-configure actions for the Reader to take at some future time. This allows time sensitive operations like triggering and tag operations, to be performed in the Reader without the need for low-latency control from the Client.

ROSpecs pass the rules for Reader operations. AISpecs pass the rules for inventory operations. AccessSpecs pass the rules for tag access operations etc.

By combining these "specs" with the LLRP start and stop commands, the LLRP protocol can also facilitate operation in imperative manner, allowing in real-time interaction between Reader and Client application.

8. What is a custom extension in LLRP?

LLRP provides all the basis functionality of a UHFC1G2 reader including access to low-level configuration. However, it was expected and anticipated by the group, that not all reader operations or behaviors could be standardized. Custom extensions provide a framework for Reader and Clients side to augment LLRP functionality rather than replace this functionality with a proprietary interface. An LLRP reader or Client does not have to create a proprietary protocol just to expose one unique enhancement—it can be done through an LLRP custom extension.

LLRP provides and a mechanism to "name" or uniquely identify a custom extension so extensions by two different parties can be easily identified and reconciled. It also indicates the legal extension points, so Reader and Client implementations can architect and plan for extensions in a structured manner.

9. LLRP is a binary protocol. What does this mean? Why is LLRP binary?

A binary protocol is one where information is sent in machine readable form as opposed to human readable form. The charter required that implementation of LLRP be suitable for performance and space-constrained applications. A binary protocol often provides this benefit.

10. Who connects to whom in LLRP? Is it a symmetric protocol?

LLRP is not a symmetric protocol. Generally, specs and commands are sent from the Client to the Reader. Reports and events are sent from the Reader to the Client.

However, either party is able to establish the LLRP connection. The working group discussed different deployment models, network stability, network topology, addressing, and scalability. For a standard to apply to the range of deployments, the group concluded that bi-directional initiation of connections was required.

11. Does LLRP accumulate, or filter tag data?

First, some quick definitions: Accumulation is the gathering of one or more tag reports, combining reports that contain duplicate data. Filtering is limiting reports to a configurable subset of EPCs.

LLRP is an interface to send messages between a client and a reader. The LLRP specification does not require that a Reader or Client accumulate or store report data. However, the protocol includes a normative set of accumulation rules, and the configuration options to trigger tag report generation.

LLRP exposes Gen2 select filtering in two different ways, allowing clients air-protocol specific control over filtering Tag populations.

LLRP accumulation and filtering do not replace traditional application level filtering of data. LLRP is defined between a single Client and single Reader and provides no support to accumulate data from multiple readers.

12. How do I reboot, reset, or upgrade firmware on LLRP readers?

LLRP offers no standard mechanism to reboot or reset a reader or to upgrade firmware.

13. What is Tag Inventory State Aware Singulation?

You are referring to the TagInventoryStateAware Boolean field in the C1G2InventoryCommand parameter and the two air-protocol specific LLRP parameters C1G2TagInventoryStateAwareFilterAction and C1G2TagInventoryStateAwareSingulationAction. When TagInventoryStateAware is set false, then the LLRP Client is letting the Reader manage tag inventory state information. When this Boolean field is set true, then the Client is taking responsibility for managing tag inventory state information.

The C1G2TagInventoryStateAwareFilterAction parameter gives the LLRP Client access to two parameters in the C1G2 air protocol Select command (Target and Action parameters, see section 6.3.2.10.1 of the C1G2 spec v1.0.9). When using this parameter it will be necessary to understand the operation of the Select command as defined by the C1G2 spec.

The C1G2TagInventoryStateAwareSingulationAction parameter gives the LLRP Client access to two parameters in the C1G2 air protocol Query command (Target and Sel parameters, see section 6.3.2.10.2.1 of the C1G2 spec v1.0.9). See the explanation provided in the LLRP spec in section 15.2.1.2.1.3 in the C1G2SingulationControl parameter description. Also, when using this parameter it will be necessary to understand the operation of the Query command as defined by the C1G2 spec.

14. The Receive Sensitivity Value has range of 0 to 128. How is the Receive Sensitivity Value calculated?

This value is expressed in dB and is the offset from the maximum sensitivity supported by the reader. For example, if the value is set to 0 dB, the receive sensitivity setting at the reader is equal to the maximum sensitivity setting (MaxSens) supported by the reader. If the value is X dB, the receive sensitivity at the reader is equal to (MaxSens + X).