



Low Level Reader Protocol (LLRP), Version 1.0

Ratified Standard

April 12, 2007

Disclaimer

EPCglobal IncTM is providing this document as a service to interested industries. This document was developed through a consensus process of interested parties.

Although efforts have been to assure that the document is correct, reliable, and technically accurate, EPCglobal Inc. makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS DOCUMENT IS CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR WILL BE SUITABLE FOR ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of this document is with the understanding that EPCglobal Inc. has no liability for any claim to the contrary, or for any damage or loss of any kind or nature.

Copyright notice

© 2006, 2007, EPCglobal Inc.

All rights reserved. Unauthorized reproduction, modification, and/or use of this document is not permitted. Requests for permission to reproduce should be addressed to epcglobal@epcglobalinc.org.

EPCglobal Inc.TM is providing this document as a service to interested industries. This document was developed through a consensus process of interested parties. Although efforts have been to assure that the document is correct, reliable, and technically accurate, EPCglobal Inc. makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS DOCUMENT IS CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR WILL BE SUITABLE FOR ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of this Document is with the understanding that EPCglobal Inc. has no liability for any claim to the contrary, or for any damage or loss of any kind or nature

Abstract

This document specifies an interface between RFID Readers and Clients. The interface protocol 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.

Audience for this document

The target audience for this specification includes:

RFID Network Infrastructure vendors

Reader vendors

EPC Middleware vendors

System integrators

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at EPCglobal. See www.epcglobalinc.org for more information.

This version of this document has been ratified by the EPCglobal Board as of April 12, 2007

Comments on this document should be sent to the EPCglobal Software Action Group Reader Operations Working Group mailing list

sag_readerop@epclinklist.epcglobalinc.org.

Contents

58	INDEX OF FIGURES.....	11
59	INDEX OF TABLES.....	12
60	1 (INFORMATIVE) GLOSSARY.....	13
61	2 INTRODUCTION.....	16
62	3 ROLE WITHIN THE EPCGLOBAL NETWORK ARCHITECTURE.....	17
63	4 TERMINOLOGY AND TYPOGRAPHICAL CONVENTIONS.....	19
64	5 OVERVIEW OF LLRP.....	19
65	5.1 TYPICAL LLRP TIMELINE	20
66	6 LLRP OPERATION.....	21
67	6.1 INVENTORY, RF SURVEY AND ACCESS OPERATIONS	21
68	6.1.1 Operation Triggers	30
69	6.1.1.1 Summary	30
70	6.1.1.2 Reader Operation Triggers.....	31
71	6.1.1.3 Access Operation Triggers.....	32
72	6.2 REPORTING, EVENT NOTIFICATION AND KEEPALIVES.....	32
73	7 MESSAGES, PARAMETERS AND FIELDS	33
74	7.1 OVERVIEW	33
75	7.1.1 Formatting Conventions.....	33
76	7.1.2 Messages.....	34
77	7.1.3 Parameters	35
78	7.1.3.1 General Parameters	35
79	7.1.3.1.1 Timestamp	35
80	7.1.3.1.1.1 UTCTimestamp Parameter	35
81	7.1.3.1.1.2 Uptime Parameter.....	35
82	7.1.4 Fields	36
83	7.1.5 Functional Grouping	36
84	7.1.6 LLRP Messages and Actions	38
85	8 CUSTOM EXTENSION	39
86	8.1 CUSTOM_MESSAGE.....	40
87	8.2 CUSTOM_PARAMETER.....	40
88	8.3 CUSTOM EXTENSION IN COMMANDS.....	40
89	8.4 CUSTOM EXTENSION IN INDIVIDUAL LLRP PARAMETERS	41
90	8.5 ALLOWABLE PARAMETER EXTENSION	41
91	9 READER DEVICE CAPABILITIES.....	42
92	9.1 MESSAGES	42
93	9.1.1 GET_READER_CAPABILITIES	42
94	9.1.2 GET_READER_CAPABILITIES_RESPONSE	42
95	9.2 PARAMETERS	43
96	9.2.1 GeneralDeviceCapabilities Parameter.....	43
97	9.2.1.1 ReceiveSensitivityTableEntry Parameter.....	44
98	9.2.1.2 PerAntennaReceiveSensitivityRange Parameter.....	44
99	9.2.1.3 PerAntennaAirProtocol Parameter.....	45
100		

101	9.2.1.4	GPIOCapabilities Parameter	45
102	9.2.2	LLRPCapabilities Parameter	45
103	9.2.3	AirProtocolLLRPCapabilities Parameter	46
104	9.2.4	RegulatoryCapabilities Parameter	47
105	9.2.4.1	UHFBandCapabilities Parameter	47
106	9.2.4.1.1	TransmitPowerLevelTableEntry Parameter	48
107	9.2.4.1.2	FrequencyInformation Parameter	48
108	9.2.4.1.2.1	FrequencyHopTable Parameter	48
109	9.2.4.1.2.2	FixedFrequencyTable Parameter	49
110	10	READER OPERATION (RO).....	49
111	10.1	MESSAGES	49
112	10.1.1	ADD_ROSPEC	49
113	10.1.2	ADD_ROSPEC_RESPONSE	50
114	10.1.3	DELETE_ROSPEC	50
115	10.1.4	DELETE_ROSPEC_RESPONSE	50
116	10.1.5	START_ROSPEC	50
117	10.1.6	START_ROSPEC_RESPONSE	51
118	10.1.7	STOP_ROSPEC	51
119	10.1.8	STOP_ROSPEC_RESPONSE	51
120	10.1.9	ENABLE_ROSPEC	52
121	10.1.10	ENABLE_ROSPEC_RESPONSE	52
122	10.1.11	DISABLE_ROSPEC	52
123	10.1.12	DISABLE_ROSPEC_RESPONSE	52
124	10.1.13	GET_ROSPECS	53
125	10.1.14	GET_ROSPECS_RESPONSE	53
126	10.2	PARAMETERS	53
127	10.2.1	ROSpec Parameter	53
128	10.2.1.1	ROBoundarySpec Parameter	54
129	10.2.1.1.1	ROSpecStartTrigger Parameter	54
130	10.2.1.1.1.1	PeriodicTriggerValue Parameter	54
131	10.2.1.1.1.2	GPITriggerValue Parameter	55
132	10.2.1.1.2	ROSpecStopTrigger Parameter	55
133	10.2.2	AISpec Parameter	56
134	10.2.2.1	AISpecStopTrigger Parameter	56
135	10.2.2.1.1	TagObservationTrigger Parameter	57
136	10.2.2.2	InventoryParameterSpec Parameter	57
137	10.2.3	RFSurveySpec Parameter	58
138	10.2.3.1	RFSurveySpecStopTrigger Parameter	58
139	11	ACCESS OPERATION.....	59
140	11.1	MESSAGES	59
141	11.1.1	ADD_ACCESSSPEC	59
142	11.1.2	ADD_ACCESSSPEC_RESPONSE	59
143	11.1.3	DELETE_ACCESSSPEC	59
144	11.1.4	DELETE_ACCESSSPEC_RESPONSE	60
145	11.1.5	ENABLE_ACCESSSPEC	60
146	11.1.6	ENABLE_ACCESSSPEC_RESPONSE	60
147	11.1.7	DISABLE_ACCESSSPEC	61
148	11.1.8	DISABLE_ACCESSSPEC_RESPONSE	61
149	11.1.9	GET_ACCESSSPECS	61
150	11.1.10	GET_ACCESSSPECS_RESPONSE	61
151	11.1.11	CLIENT_REQUEST_OP	62
152	11.1.12	CLIENT_REQUEST_OP_RESPONSE	62
153	11.2	PARAMETERS	62
154	11.2.1	AccessSpec Parameter	62
155	11.2.1.1	AccessSpecStopTrigger Parameter	63

156	11.2.1.2	AccessCommand Parameter.....	63
157	11.2.1.2.1	ClientRequestOpSpec Parameter	64
158	11.2.2	ClientRequestResponse Parameter	65
159	12	READER DEVICE CONFIGURATION	65
160	12.1	MESSAGES	65
161	12.1.1	GET_READER_CONFIG.....	65
162	12.1.2	GET_READER_CONFIG_RESPONSE.....	66
163	12.1.3	SET_READER_CONFIG.....	67
164	12.1.4	SET_READER_CONFIG_RESPONSE.....	68
165	12.1.5	CLOSE_CONNECTION.....	68
166	12.1.6	CLOSE_CONNECTION_RESPONSE.....	69
167	12.2	PARAMETERS	69
168	12.2.1	LLRPConfigurationStateValue Parameter	69
169	12.2.2	Identification Parameter.....	70
170	12.2.3	GPOWriteData Parameter	70
171	12.2.4	KeepaliveSpec Parameter.....	71
172	12.2.5	AntennaProperties Parameter	71
173	12.2.6	AntennaConfiguration Parameter.....	72
174	12.2.6.1	RFReceiver Parameter	72
175	12.2.6.2	RFTransmitter Parameter.....	72
176	12.2.6.3	GPIOPortCurrentState Parameter	73
177	12.2.6.4	EventsAndReports Parameter	73
178	13	REPORTS, NOTIFICATIONS AND KEEPALIVES	74
179	13.1	MESSAGES	74
180	13.1.1	GET_REPORT.....	74
181	13.1.2	RO_ACCESS_REPORT.....	74
182	13.1.3	KEEPALIVE.....	75
183	13.1.4	KEEPALIVE_ACK.....	75
184	13.1.5	READER_EVENT_NOTIFICATION.....	75
185	13.1.6	ENABLE_EVENTS_AND_REPORTS.....	75
186	13.2	PARAMETERS	76
187	13.2.1	ROReportSpec Parameter.....	76
188	13.2.1.1	TagReportContentSelector Parameter.....	77
189	13.2.2	AccessReportSpec Parameter	77
190	13.2.3	TagReportData Parameter	78
191	13.2.3.1	Accumulation of TagReportData	79
192	13.2.3.2	EPCData Parameter.....	80
193	13.2.3.3	ROSpecID Parameter.....	80
194	13.2.3.4	SpecIndex Parameter.....	80
195	13.2.3.5	InventoryParameterSpecID Parameter	80
196	13.2.3.6	AntennaID Parameter.....	81
197	13.2.3.7	PeakRSSI Parameter	81
198	13.2.3.8	ChannelIndex Parameter	81
199	13.2.3.9	FirstSeenTimestampUTC Parameter.....	81
200	13.2.3.10	FirstSeenTimestampUptime Parameter.....	82
201	13.2.3.11	LastSeenTimestampUTC Parameter	82
202	13.2.3.12	LastSeenTimestampUptime Parameter	82
203	13.2.3.13	TagSeenCount Parameter.....	82
204	13.2.3.14	ClientRequestOpSpecResult Parameter	82
205	13.2.3.15	AccessSpecID Parameter	83
206	13.2.4	RFSurveyReportData Parameter	83
207	13.2.4.1	FrequencyRSSILevelEntry Parameter	83
208	13.2.5	ReaderEventNotificationSpec Parameter	84
209	13.2.5.1	EventNotificationState Parameter	84
210	13.2.6	ReaderEventNotificationData Parameter.....	84
211	13.2.6.1	Requirements for Ordering of Event Reporting	85

212	13.2.6.2	HoppingEvent Parameter	86
213	13.2.6.3	GPISpecEvent Parameter	86
214	13.2.6.4	ROSpecEvent Parameter	86
215	13.2.6.5	ReportBufferLevelWarningEvent Parameter	87
216	13.2.6.6	ReportBufferOverflowErrorEvent Parameter	87
217	13.2.6.7	ReaderExceptionEvent Parameter	88
218	13.2.6.7.1	OpSpecID Parameter	88
219	13.2.6.8	RFSurveyEvent Parameter	88
220	13.2.6.9	AIISpecEvent Parameter	89
221	13.2.6.10	AntennaEvent Parameter	89
222	13.2.6.11	ConnectionAttemptEvent Parameter	90
223	13.2.6.12	ConnectionCloseEvent Parameter	90
224	14	ERRORS	90
225	14.1	MESSAGES	91
226	14.1.1	ERROR_MESSAGE	91
227	14.2	PARAMETERS	91
228	14.2.1	LLRP Status Codes	91
229	14.2.2	LLRPStatus Parameter	93
230	14.2.2.1	FieldError Parameter	94
231	14.2.2.2	ParameterError Parameter	94
232	15	AIR PROTOCOL SPECIFIC PARAMETERS	95
233	15.1	LLRP AIR PROTOCOL CROSS-REFERENCE TABLES	95
234	15.1.1	Class-1 Generation-2 (C1G2) Air Protocol	95
235	15.2	LLRP AIR PROTOCOL SPECIFIC PARAMETERS	96
236	15.2.1	Class-1 Generation-2 (C1G2) Air Protocol	96
237	15.2.1.1	Reader Device Capabilities	96
238	15.2.1.1.1	C1G2LLRPCapabilities Parameter	96
239	15.2.1.1.2	UHFC1G2RFModeTable Parameter	96
240	15.2.1.1.2.1	UHFC1G2RFModeTableEntry Parameter	97
241	15.2.1.2	Inventory Operation	98
242	15.2.1.2.1	C1G2InventoryCommand Parameter	98
243	15.2.1.2.1.1	C1G2Filter Parameter	99
244	15.2.1.2.1.1.1	C1G2TagInventoryMask Parameter	100
245	15.2.1.2.1.1.2	C1G2TagInventoryStateAwareFilterAction Parameter	100
246	15.2.1.2.1.1.3	C1G2TagInventoryStateUnawareFilterAction Parameter	101
247	15.2.1.2.1.2	C1G2RF Control Parameter	102
248	15.2.1.2.1.3	C1G2SingulationControl Parameter	102
249	15.2.1.2.1.3.1	C1G2TagInventoryStateAwareSingulationAction Parameter	103
250	15.2.1.3	Access Operation	104
251	15.2.1.3.1	C1G2TagSpec Parameter	104
252	15.2.1.3.1.1	C1G2TargetTag Parameter	104
253	15.2.1.3.2	C1G2 OpSpec Parameters	105
254	15.2.1.3.2.1	C1G2Read Parameter	105
255	15.2.1.3.2.2	C1G2Write Parameter	106
256	15.2.1.3.2.3	C1G2Kill Parameter	106
257	15.2.1.3.2.4	C1G2Lock Parameter	106
258	15.2.1.3.2.4.1	C1G2LockPayload Parameter	107
259	15.2.1.3.2.5	C1G2BlockErase Parameter	107
260	15.2.1.3.2.6	C1G2BlockWrite Parameter	108
261	15.2.1.4	Reader Device Configuration	108
262	15.2.1.5	Reports	108
263	15.2.1.5.1	C1G2EPCMemorySelector Parameter	108
264	15.2.1.5.2	C1G2PC Parameter	109
265	15.2.1.5.3	C1G2CRC Parameter	109
266	15.2.1.5.4	C1G2SingulationDetails Parameter	109
267	15.2.1.5.5	C1G2 OpSpec Results	109
268	15.2.1.5.5.1	C1G2ReadOpSpecResult Parameter	109
269	15.2.1.5.5.2	C1G2WriteOpSpecResult Parameter	110
270	15.2.1.5.5.3	C1G2KillOpSpecResult Parameter	110

271	15.2.1.5.5.4	C1G2LockOpSpecResult Parameter.....	111
272	15.2.1.5.5.5	C1G2BlockEraseOpSpecResult Parameter.....	111
273	15.2.1.5.5.6	C1G2BlockWriteOpSpecResult Parameter	111
274	16	BINARY ENCODING FOR LLRP	112
275	16.1	MESSAGES	113
276	16.1.1	GET_READER_CAPABILITIES.....	115
277	16.1.2	GET_READER_CAPABILITIES_RESPONSE	115
278	16.1.3	ADD_ROSPEC	115
279	16.1.4	ADD_ROSPEC_RESPONSE	116
280	16.1.5	DELETE_ROSPEC.....	116
281	16.1.6	DELETE_ROSPEC_RESPONSE.....	116
282	16.1.7	START_ROSPEC	116
283	16.1.8	START_ROSPEC_RESPONSE	117
284	16.1.9	STOP_ROSPEC.....	117
285	16.1.10	STOP_ROSPEC_RESPONSE.....	117
286	16.1.11	ENABLE_ROSPEC.....	117
287	16.1.12	ENABLE_ROSPEC_RESPONSE.....	118
288	16.1.13	DISABLE_ROSPEC.....	118
289	16.1.14	DISABLE_ROSPEC_RESPONSE.....	118
290	16.1.15	GET_ROSPECS.....	118
291	16.1.16	GET_ROSPECS_RESPONSE.....	119
292	16.1.17	ADD_ACCESSSPEC.....	119
293	16.1.18	ADD_ACCESSSPEC_RESPONSE	119
294	16.1.19	DELETE_ACCESSSPEC.....	120
295	16.1.20	DELETE_ACCESSSPEC_RESPONSE.....	120
296	16.1.21	ENABLE_ACCESSSPEC.....	120
297	16.1.22	ENABLE_ACCESSSPEC_RESPONSE.....	120
298	16.1.23	DISABLE_ACCESSSPEC.....	121
299	16.1.24	DISABLE_ACCESSSPEC_RESPONSE.....	121
300	16.1.25	GET_ACCESSSPECS	121
301	16.1.26	GET_ACCESSSPECS_RESPONSE	121
302	16.1.27	CLIENT_REQUEST_OP	122
303	16.1.28	CLIENT_REQUEST_OP_RESPONSE	122
304	16.1.29	GET_REPORT.....	122
305	16.1.30	RO_ACCESS_REPORT	123
306	16.1.31	KEEPALIVE.....	123
307	16.1.32	KEEPALIVE_ACK.....	123
308	16.1.33	READER_EVENT_NOTIFICATION	124
309	16.1.34	ENABLE_EVENTS_AND_REPORTS.....	124
310	16.1.35	ERROR_MESSAGE	124
311	16.1.36	GET_READER_CONFIG.....	124
312	16.1.37	GET_READER_CONFIG_RESPONSE.....	125
313	16.1.38	SET_READER_CONFIG.....	125
314	16.1.39	SET_READER_CONFIG_RESPONSE.....	126
315	16.1.40	CLOSE_CONNECTION	126
316	16.1.41	CLOSE_CONNECTION_RESPONSE.....	126
317	16.1.42	CUSTOM_MESSAGE	127
318	16.2	LLRP PARAMETERS.....	127
319	16.2.1	TLV and TV Encoding of LLRP Parameter	127
320	16.2.1.1	TLV-Parameters.....	127
321	16.2.1.1.1	Encoding Guidelines for TLV-Parameters.....	128
322	16.2.1.2	TV-Parameters	128
323	16.2.1.2.1	Encoding Guidelines for TV-Parameters	129
324	16.2.2	General Parameters.....	129
325	16.2.2.1	UTCTimestamp Parameter.....	129
326	16.2.2.2	Uptime Parameter	129

327	16.2.3	<i>Reader Device Capabilities Parameters</i>	129
328	16.2.3.1	GeneralDeviceCapabilities Parameter.....	129
329	16.2.3.1.1	ReceiveSensitivityTableEntry Parameter.....	130
330	16.2.3.1.2	PerAntennaReceiveSensitivityRange Parameter.....	130
331	16.2.3.1.3	PerAntennaAirProtocol Parameter.....	130
332	16.2.3.1.4	GPiOCapabilities Parameter.....	131
333	16.2.3.2	LLRPCapabilities Parameter.....	131
334	16.2.3.3	AirProtocolLLRPCapabilities Parameter.....	131
335	16.2.3.4	RegulatoryCapabilities Parameter.....	131
336	16.2.3.4.1	UHFBandCapabilities Parameter.....	132
337	16.2.3.4.1.1	TransmitPowerLevelTableEntry Parameter.....	132
338	16.2.3.4.1.2	FrequencyInformation Parameter.....	132
339	16.2.3.4.1.2.1	FrequencyHopTable Parameter.....	132
340	16.2.3.4.1.2.2	FixedFrequencyTable Parameter.....	133
341	16.2.4	<i>Reader Operations Parameters</i>	133
342	16.2.4.1	ROSpec Parameter.....	133
343	16.2.4.1.1	ROBoundarySpec Parameter.....	133
344	16.2.4.1.1.1	ROSpecStartTrigger Parameter.....	134
345	16.2.4.1.1.1.1	PeriodicTriggerValue Parameter.....	134
346	16.2.4.1.1.1.2	GPITriggerValue Parameter.....	134
347	16.2.4.1.1.2	ROSpecStopTrigger Parameter.....	134
348	16.2.4.2	AISpec Parameter.....	135
349	16.2.4.2.1	AISpecStopTrigger Parameter.....	135
350	16.2.4.2.1.1	TagObservationTrigger Parameter.....	135
351	16.2.4.2.2	InventoryParameterSpec Parameter.....	135
352	16.2.4.3	RFSurveySpec Parameter.....	136
353	16.2.4.3.1	RFSurveySpecStopTrigger Parameter.....	136
354	16.2.5	<i>Access Operation Parameters</i>	136
355	16.2.5.1	AccessSpec Parameter.....	136
356	16.2.5.1.1	AccessSpecStopTrigger Parameter.....	137
357	16.2.5.1.2	AccessCommand Parameter.....	137
358	16.2.5.1.3	ClientRequestOpSpec Parameter.....	137
359	16.2.5.1.3.1	ClientRequestResponse Parameter.....	137
360	16.2.6	<i>Configuration Parameters</i>	138
361	16.2.6.1	LLRPConfigurationStateValue Parameter.....	138
362	16.2.6.2	Identification Parameter.....	138
363	16.2.6.3	GPOWriteData Parameter.....	138
364	16.2.6.4	KeepaliveSpec Parameter.....	138
365	16.2.6.5	AntennaProperties Parameter.....	139
366	16.2.6.6	AntennaConfiguration Parameter.....	139
367	16.2.6.7	RFReceiver Parameter.....	139
368	16.2.6.8	RFTransmitter Parameter.....	139
369	16.2.6.9	GPiOPortCurrentState Parameter.....	140
370	16.2.6.10	EventsAndReports Parameter.....	140
371	16.2.7	<i>Reporting Parameters</i>	140
372	16.2.7.1	ROReportSpec Parameter.....	140
373	16.2.7.1.1	TagReportContentSelector Parameter.....	140
374	16.2.7.2	AccessReportSpec Parameter.....	141
375	16.2.7.3	TagReportData Parameter.....	141
376	16.2.7.3.1	EPCData Parameter.....	142
377	16.2.7.3.2	EPC-96 Parameter (TV-Encoding).....	142
378	16.2.7.3.3	ROSpecID Parameter (TV-Encoding).....	142
379	16.2.7.3.4	SpecIndex Parameter (TV-Encoding).....	143
380	16.2.7.3.5	InventoryParameterSpecID Parameter (TV-Encoding).....	143
381	16.2.7.3.6	AntennaID Parameter (TV-Encoding).....	143
382	16.2.7.3.7	PeakRSSI Parameter (TV-Encoding).....	143
383	16.2.7.3.8	ChannelIndex Parameter (TV-Encoding).....	143
384	16.2.7.3.9	FirstSeenTimestampUTC Parameter (TV-Encoding).....	143
385	16.2.7.3.10	FirstSeenTimestampUptime Parameter (TV-Encoding).....	143
386	16.2.7.3.11	LastSeenTimestampUTC Parameter (TV-Encoding).....	144
387	16.2.7.3.12	LastSeenTimestampUptime Parameter (TV-Encoding).....	144
388	16.2.7.3.13	TagSeenCount Parameter (TV-Encoding).....	144

389	16.2.7.3.14	ClientRequestOpSpecResult Parameter (TV-Encoding).....	144
390	16.2.7.3.15	AccessSpecID Parameter (TV-Encoding).....	144
391	16.2.7.4	RFSurveyReportData Parameter.....	145
392	16.2.7.4.1	FrequencyRSSIlevelEntry Parameter.....	145
393	16.2.7.5	ReaderEventNotificationSpec Parameter.....	145
394	16.2.7.5.1	EventNotificationState Parameter.....	146
395	16.2.7.6	ReaderEventNotificationData Parameter.....	146
396	16.2.7.6.1	HoppingEvent Parameter.....	146
397	16.2.7.6.2	GPiEvent Parameter.....	147
398	16.2.7.6.3	ROSpecEvent Parameter.....	147
399	16.2.7.6.4	ReportBufferLevelWarningEvent Parameter.....	147
400	16.2.7.6.5	ReportBufferOverflowErrorEvent Parameter.....	147
401	16.2.7.6.6	ReaderExceptionEvent Parameter.....	147
402	16.2.7.6.6.1	OpSpecID Parameter (TV-Encoding).....	148
403	16.2.7.6.7	RFSurveyEvent Parameter.....	148
404	16.2.7.6.8	AI SpecEvent Parameter.....	148
405	16.2.7.6.9	AntennaEvent Parameter.....	148
406	16.2.7.6.10	ConnectionAttemptEvent Parameter.....	149
407	16.2.7.6.11	ConnectionCloseEvent Parameter.....	149
408	16.2.8	LLRP Error Parameters.....	149
409	16.2.8.1	LLRPStatus Parameter.....	149
410	16.2.8.1.1	FieldError Parameter.....	149
411	16.2.8.1.2	ParameterError Parameter.....	150
412	16.2.9	Custom Parameter.....	150
413	16.3	AIR PROTOCOL SPECIFIC PARAMETERS.....	150
414	16.3.1	Class-1 Generation-2 (C1G2) Protocol Parameters.....	150
415	16.3.1.1	Capabilities Parameters.....	151
416	16.3.1.1.1	C1G2LLRPCapabilities Parameter.....	151
417	16.3.1.1.2	UHFC1G2RFModeTable Parameter.....	151
418	16.3.1.1.2.1	UHFC1G2RFModeTableEntry Parameter.....	151
419	16.3.1.2	Reader Operations Parameters.....	152
420	16.3.1.2.1	C1G2InventoryCommand Parameter.....	152
421	16.3.1.2.1.1	C1G2Filter Parameter.....	152
422	16.3.1.2.1.1.1	C1G2TagInventoryMask Parameter.....	152
423	16.3.1.2.1.1.2	C1G2TagInventoryStateAwareFilterAction Parameter.....	153
424	16.3.1.2.1.1.3	C1G2TagInventoryStateUnawareFilterAction Parameter.....	153
425	16.3.1.2.1.2	C1G2RFControl Parameter.....	153
426	16.3.1.2.1.3	C1G2SingulationControl Parameter.....	153
427	16.3.1.2.1.3.1	C1G2TagInventoryStateAwareSingulationAction Parameter.....	154
428	16.3.1.3	Access Operation Parameters.....	154
429	16.3.1.3.1	C1G2TagSpec Parameter.....	154
430	16.3.1.3.1.1	C1G2TargetTag Parameter.....	154
431	16.3.1.3.2	C1G2 OpSpecs.....	154
432	16.3.1.3.2.1	C1G2Read Parameter.....	155
433	16.3.1.3.2.2	C1G2Write Parameter.....	155
434	16.3.1.3.2.3	C1G2Kill Parameter.....	155
435	16.3.1.3.2.4	C1G2Lock Parameter.....	155
436	16.3.1.3.2.4.1	C1G2LockPayload Parameter.....	156
437	16.3.1.3.2.5	C1G2BlockErase Parameter.....	156
438	16.3.1.3.2.6	C1G2BlockWrite Parameter.....	156
439	16.3.1.4	Configuration Parameters.....	156
440	16.3.1.5	Reporting Parameters.....	157
441	16.3.1.5.1	C1G2EPCMemorySelector Parameter.....	157
442	16.3.1.5.2	C1G2PC Parameter (TV-Encoding).....	157
443	16.3.1.5.3	C1G2CRC Parameter (TV-Encoding).....	157
444	16.3.1.5.4	C1G2SingulationDetails Parameter (TV-Encoding).....	157
445	16.3.1.5.5	C1G2 OpSpec Results.....	157
446	16.3.1.5.5.1	C1G2ReadOpSpecResult Parameter.....	157
447	16.3.1.5.5.2	C1G2WriteOpSpecResult Parameter.....	158
448	16.3.1.5.5.3	C1G2KillOpSpecResult Parameter.....	158
449	16.3.1.5.5.4	C1G2LockOpSpecResult Parameter.....	158
450	16.3.1.5.5.5	C1G2BlockEraseOpSpecResult Parameter.....	158

451	16.3.1.5.5.6	C1G2BlockWriteOpSpecResult Parameter	159
452	16.4	LISTING OF MESSAGE AND PARAMETER TYPES.....	159
453	17	TRANSMITTER BEHAVIOR OF A READER.....	163
454	18	CONNECTION AND TRANSPORT	164
455	18.1	TCP TRANSPORT.....	164
456	18.2	SECURITY IN TCP TRANSPORT.....	171
457	18.2.1	<i>Normative Section</i>	171
458	18.2.2	<i>Informative Section</i>	171
459	18.2.2.1	Overview of TLS	171
460	18.2.2.2	Threat Analysis for LLRP.....	172
461	18.2.2.3	Configuration Elements for TLS.....	172
462	18.2.2.4	Why different TLS server port?	173
463	19	(INFORMATIVE) OBJECT MODEL.....	173
464	19.1	CAPABILITIES	175
465	19.2	CONFIGURATION	176
466	19.3	ROSpec	178
467	19.4	ACCESSSPEC	179
468	19.5	C1G2 PARAMETERS	180
469	19.6	REPORTING AND NOTIFICATION	181
470	19.7	GENERAL	184
471	20	(INFORMATIVE) TCP KEEPALIVES	184
472	21	(INFORMATIVE) REFERENCES.....	185
473	22	ACKNOWLEDGEMENT OF CONTRIBUTORS AND COMPANIES OPT'D-IN DURING	
474		THE CREATION OF THIS STANDARD (INFORMATIVE)	187
475			
476			

477 Index of Figures

478	FIGURE 1: LLRP IN THE EPCGLOBAL ARCHITECTURE	17
479	FIGURE 2: LLRP ENDPOINTS.....	20
480	FIGURE 3: TYPICAL LLRP TIMELINE.....	21
481	FIGURE 4: ROSPEC STATECHART	23
482	FIGURE 5: ANTENNA INVENTORY SPEC STATES	24
483	FIGURE 6: ACCESS OPERATIONS INTERLEAVED IN AN ANTENNA INVENTORY OPERATION	25
484	FIGURE 7: CLIENT REQUEST OpSPEC	26
485	FIGURE 8: ACCESS SPEC STATES.....	27
486	FIGURE 9: RFSURVEYSPEC STATES	28
487	FIGURE 10: BOX FORMATS FOR MESSAGES AND PARAMETERS.....	33
488	FIGURE 11: LLRP MESSAGES AND READER ACTIONS.....	39
489	FIGURE 12: NETWORK ORDER.....	112
490	FIGURE 13: BIT ORDER IN FIELDS	113
491	FIGURE 14: READER INITIATED CONNECTION (NORMAL)	165
492	FIGURE 15: READER INITIATED CONNECTION (EXCEPTION).....	166
493	FIGURE 16: CLIENT INITIATED CONNECTION (NORMAL).....	166
494	FIGURE 17: CLIENT INITIATED CONNECTION (EXCEPTION #1)	167
495	FIGURE 18: CLIENT INITIATED CONNECTION (EXCEPTION #2).....	167
496	FIGURE 19: CLIENT INITIATED CONNECTION (EXCEPTION #3).....	168
497	FIGURE 20: CLIENT INITIATED CONNECTION (EXCEPTION #4).....	168
498	FIGURE 21: CLIENT INITIATED CONNECTION (EXCEPTION #5).....	169
499	FIGURE 22: READER INITIATED CLOSE.....	169
500	FIGURE 23: CLIENT INITIATED CLOSE (NORMAL)	170
501	FIGURE 24: CLIENT INITIATED CLOSE (EXCEPTION).....	170
502	FIGURE 25: CAPABILITIES	175
503	FIGURE 26: CONFIGURATION.....	176
504	FIGURE 27: CONFIGURATION COMMANDS.....	177
505	FIGURE 28: RO COMMANDS.....	178
506	FIGURE 29: ROSPEC.....	178
507	FIGURE 30: ACCESS COMMANDS.....	179
508	FIGURE 31: ACCESSPEC.....	179
509	FIGURE 32: C1G2 PARAMETERS.....	180
510	FIGURE 33: C1G2 INVENTORY COMMAND.....	180
511	FIGURE 34: C1G2 ACCESSPEC	181
512	FIGURE 35: REPORTING AND NOTIFICATION	181
513	FIGURE 36: TAGREPORTDATA	182
514	FIGURE 37: RFSURVEYREPORTDATA	183
515	FIGURE 38: READER EVENT NOTIFICATION DATA	183
516	FIGURE 39: GENERAL DATA.....	184
517		

518	Index of Tables	
519	TABLE 1: OPERATION TRIGGERS	30
520	TABLE 2: REPORTING TRIGGERS	32
521	TABLE 3: AIR PROTOCOL ENUMERATIONS USED IN LLRP	36
522	TABLE 4: MESSAGE LISTING	159
523	TABLE 5: PARAMETER LISTING	160
524		

525 1 (Informative) Glossary

526 This section provides a non-normative summary of terms used within this specification.

Term	Meaning
Access	The operation of communicating with (reading from and/or writing to) a Tag. An individual Tag must be uniquely identified prior to access. Access comprises multiple commands, some of which employ one-time-pad based cover-coding of the R=>T link.
Air Interface	The complete communication link between a Reader and a Tag including the physical layer, collision arbitration algorithm, command and response structure, and data-coding methodology.
Antenna	An atomic, specifically-addressable RF transmission and/or reception device used for communication with RFID tags. For the purposes of this spec, multiplicity of antenna is going to be referred to as antennas.
Capabilities	The set of intrinsic Reader properties relevant to protocol operation. This may include physical, functional, or protocol support information.
Compatibility	A general term used to describe a consistency of terminology and/or operation between one or more specifications and/or implementations. It is specifically not intended to be used to define expectations on protocol operation. The proper term for this is 'Interoperability' as defined below.
Configuration	Data and parameters to control specific operation of a Reader or the Client that is typically instantiated at boot time or as a result of specific management actions on a timescale much greater than the operations of LLRP. It is possible that certain parameters may be controlled via LLRP and have corresponding default configuration parameters.
Client	From the perspective of LLRP, a <i>Client</i> is synonymous with <i>Controller</i> (see below). The specification uses the term Client to identify the endpoint opposite to the Reader.
Controller	The function that implements the Reader Protocol (Interface) opposite the Reader (i.e., an LLRP endpoint). In the EPCGlobal Architecture, this function could comprise part of the Filtering & Collection (Role), but it may be implemented in a wide range of devices, including dedicated RFID infrastructure, Readers, and middleware running on server hardware.

Term	Meaning
GPI	General purpose input
GPO	General purpose output
Interference	There are two types of interference that impact a RFID system's operating capacity: Reader-to-tag, and Reader-to-Reader. Reader-to-tag interference happens when a tag receives signals of comparable strengths from more than one Reader at the same time. This causes the tag to respond arbitrarily to the Readers, and makes its state unpredictable. Reader-to-Reader interference happens when a Reader in the midst of listening to a tag's reply at a particular frequency, receives signals much stronger than the tag's reply, from another Reader operating at the same frequency at the same time. This causes the Reader's receiver logic to not be able to correctly decode the tag's reply. Both these interference scenarios can potentially degrade the system performance.
Interoperability	The ability for two implementations of protocol endpoints to properly function with each other. Proper function may require negotiation of supported capabilities between the two endpoints.
Interrogator	Synonymous with Reader. The EPCglobal Class-1 Gen-2 air protocol specification refers to Readers as Interrogators. However, since the term Reader is included in the title of this specification <i>Low Level Reader Protocol</i> , the term Reader is used instead of Interrogator.
Inventory	The operation of identifying Tags. A Reader begins an inventory round by transmitting a Query command (Query starts the round) in one of four sessions. One or more Tags may reply. The Reader detects a single Tag reply and requests the PC, EPC, and CRC-16 from the Tag. Inventory comprises multiple commands. An inventory round operates in one and only one session (defined below) at a time.
LLRP	Low Level Reader Protocol
LLRP connection	Instance of LLRP between the Reader and the Client.
LLRP endpoint	The endpoints of a LLRP instance (i.e., either a Reader or a Client).

Term	Meaning
Q	A parameter that a Reader uses to regulate the probability of Tag response. A Reader commands Tags in an inventory round to load a Q-bit random (or pseudo-random) number into their slot counter; the Reader may also command Tags to decrement their slot counter. Tags reply when the value in their slot counter is zero. Q is an integer in the range (0,15); the corresponding Tag-response probabilities range from $2^0 = 1$ to $2^{-15} = 0.000031$.
Q algorithm	A collision-arbitration algorithm where Tags load a random (or pseudo-random) number into a slot counter, decrement this slot counter based on Reader commands, and reply to the Reader when their slot counter reaches zero.
Reader	The function that implements the RFID Reader (Role) in the EPCGlobal Architecture Specification. It is one of the two endpoints of the Reader Protocol (Interface) which is, for the purposes of this specification, LLRP. The Reader comprises of one or more antennas which are used to communicate with RFID tags. Note that a Reader can not only read RFID tags, it can perform other operations on tags such as write and kill.
Receive Sensitivity	Receiver sensitivity is a measure of the weakest tag signal an RFID reader is able to detect and demodulate. Changing this affects the minimum detectable signal (MDS) so as to prevent weaker responses from tying up the receiver. The other commonly used term for such a control is squelch.
Select	The operation of choosing a tag population for inventory and access. A Select command may be applied successively to select a particular Tag population based on user-specified criteria. This operation is analogous to selecting records from a database.
Session	An inventory process comprising a Reader and an associated Tag population. A Reader chooses one of four sessions and inventories Tags within that session. The Reader and associated Tag population operate in one and only one session for the duration of an inventory round. For each session, Tags maintain a corresponding inventoried flag. Sessions allow Tags to keep track of their inventoried status separately for each of four possible time-interleaved inventory processes, using an independent inventoried flag for each process.
Singulation	Identifying an individual Tag in a multiple-Tag environment.
Spec	The document uses the term 'Spec' to denote the parameter specification for an operation.

Term	Meaning
UTC	Coordinated Universal Time (UTC) is the international time standard as maintained by the Bureau International des Poids et Mesures (BIPM).

2 Introduction

This document specifies an interface between RFID Readers and Clients. 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. The interface described herein, and the functionality it implies, is called “Low Level Reader Protocol,” or LLRP.

Following are the responsibilities of this interface:

- Provide means to command an RFID Reader to inventory tags (read the EPC codes carried on tags), read tags (read other data on the tags apart from the EPC code), write tags, and execute other protocol-dependent access commands (such as ‘kill’ and ‘lock’ from EPCglobal Class 1 Generation 2).
- Provide means for robust status reporting and error handling during tag access operations.
- Provide means for conveying tag passwords necessary to effect commands that may require them, such as the ‘Kill’ command in the EPCglobal Class 1 Generation 2 UHF Air Interface Protocol.
- Provide means to control the forward and reverse RF link operation to manage RF power levels and spectrum utilization, and assess RF interference, among RFID Readers in a system.
- Provide means to control aspects of Tag Protocol operation, including protocol parameters and singulation parameters.
- Provide means to facilitate the addition of support for new air protocols.
- Provide means for the retrieval of Reader device capabilities.
- Provide means for vendors of Reader devices to define vendor-specific extensions to the protocol in a manner that is non-interfering among vendors, and which, to the extent possible, is vendor-administered.

In addition LLRP is “regulatory requirements-aware,” such that its functions are applicable in regulatory jurisdictions worldwide.

The overall organization of this specification is as follows: - General Overview (sections 3-6); Abstract Model (sections 7-15, 17), which describes the protocol, its message types and contents without specifying the protocol syntax; Binary Encoding (section 16), which specifies the syntax for representing the abstract protocol; Transport Binding (section 18), which specifies the mechanism for delivery of protocol messages; Informative Descriptions (sections 19-21). Guidelines for adding support of a new air protocol to LLRP are presented in section 15.1.

3 Role within the EPCglobal Network Architecture

The RFID infrastructure consists of network elements that participate in the management (e.g., read/write/lock) and transmission of tag data. The consumers of the tag data are the Client network elements (e.g., end-user applications). The network elements between the tag and the Clients form the conduit to transport tag data over the network to the applications, and convey tag operational commands over the network to the tags. The EPCGlobal Architecture (ARC) framework has outlined the roles and the associated functions performed by the various elements in this network. The elements relevant to the LLRP specification are the Tags, Readers and F&C Role.

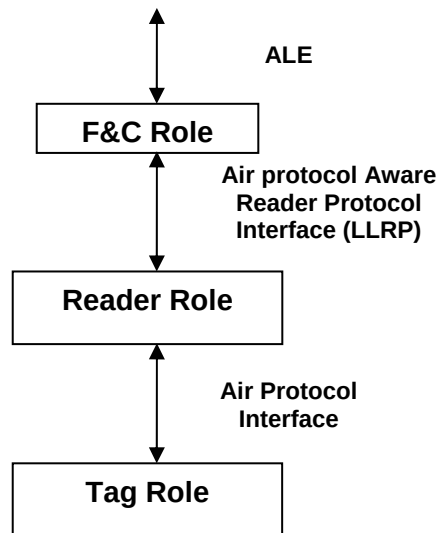


Figure 1: LLRP in the EPCGlobal Architecture

Figure 1 illustrates the position of LLRP in the EPCGlobal architecture stack between the F&C role and the Tag role.

The responsibilities of the elements and interfaces below the F&C role can be classified into three broad functional groups: tag data processing (*Data path*), Reader device management (*Management path*) and Reader control and coordination (*Control path*).

With the advent of sophisticated air protocols like UHF Class-1 Gen-2, and deployments of larger numbers of Readers, the need for Reader control and coordination (*Control path*) of the network of Readers in the architecture becomes important. The LLRP interface facilitates the control path function by exposing air protocol relevant control

knobs to the F&C role. To that effect, LLRP is designed to be extensible in terms of supporting multiple air protocols.

The physical and logical requirements for the communication between the Reader and the tag are defined by the air protocol. Specifically, the air protocol defines the signaling layer of the communication link, the Reader and tag operating procedures and commands, and the collision arbitration (also known as singulation) scheme to identify a specific tag in a multiple-tag environment. One such air protocol is the EPCGlobal Class-1 Generation 2 (C1G2) protocol. The tag memory in the C1G2 protocol is logically separated into four distinct banks: reserved memory, EPC memory, TID memory and user memory. The physical memory map of the tag is vendor-specific. The air protocol commands that access memory have a parameter that selects the bank, and an address parameter to select a particular memory location within that bank.

The fundamental operations a Reader performs on a tag population are inventory and access. Inventory is the operation of identifying tags, and comprises multiple air protocol commands. Using the singulation scheme, the Reader detects a single tag reply and requests the EPC memory contents from the tag. Access is the operation of communicating with (reading from and/or writing to) a tag. An individual tag must be uniquely identified prior to access. Similar to the inventory operation, access comprises multiple air protocol commands. In addition, a Reader can choose a subset of the tag population for inventory and access. This operation is called Select in the C1G2 protocol. The select operation is used to select and/or de-select a particular tag population for the subsequent inventory and/or access operation. This helps focus the operations on the desired subset of tags, and also thins the tag population participating in the singulation operation, thereby improving the overall singulation rate.

It is anticipated that overall system performance may be optimized by tuning the RF, singulation and air protocol parameters within and across Readers. The performance can be further optimized if the tuning is done cognizant of the RF environment in the vicinity of the Reader.

The LLRP interface between the Client and the Reader facilitates the management of Reader devices to mitigate Reader-to-tag and Reader-to-Reader interference and maximize the efficiency of singulation and data operations over the tag population. This is achieved by enabling the Reader device operation at the full performance level of the air protocol. In addition, LLRP provides the interface to transport the results of RF monitoring (a.k.a RF survey) if the Reader device is capable of performing that function.

In addition, there will be a number of applications that perform operations on the RFID tag data. Operations may range from reading EPC IDs to performing other tag access operations exposed by the air protocol like read, write, kill, lock, etc. Multiple application requirements translate into a set of access operations that a Reader or a set of Readers perform on tags as and when they are in the field of view. The LLRP interface provides a scalable mechanism to manage the access operations at the Reader devices.

Lastly, scalable device management capabilities are critical for operations of a large network of Reader devices. The LLRP interface facilitates device status and error reporting, and device capabilities discovery.

4 Terminology and Typographical Conventions

Within this specification, the terms SHALL, SHALL NOT, SHOULD, SHOULD NOT, MAY, NEED NOT, CAN, and CANNOT are to be interpreted as specified in Annex G of the ISO/IEC Directives, Part 2, 2001, 4th edition [ISODir2]. When used in this way, these terms will always be shown in ALL CAPS; when these words appear in ordinary typeface they are intended to have their ordinary English meaning. However in this document only a subset of the terms listed above SHALL be used. The subset of acceptable terms includes the following: SHALL, SHALL NOT and MAY. The terms SHOULD, SHOULD NOT, NEED NOT, CAN, and CANNOT, SHALL NOT be used.

All sections of this document, with the exception of section 1-3, 19-21, are normative, except where explicitly noted as non-normative. All figures within the document are non-normative unless otherwise specified.

The following typographical conventions are used throughout the document:

ALL CAPS type is used for the special terms from [ISODir2] enumerated above.

ALL_CAPS_UNDERSCORE type is used for LLRP message names.

CamelBackType is used for LLRP parameter and data field names.

Monospace type is used to denote programming language, UML, and XML identifiers, as well as for the text of XML documents.

5 Overview of LLRP

LLRP is specifically concerned with providing the formats and procedures of communications between a Client and a Reader. The LLRP protocol data units are called messages. Messages from the Client to the Reader include getting and setting configuration of Readers, capabilities discovery of Readers and managing the inventory and access operations at the Readers. Messages from the Reader to the Client include the reporting of Reader status, RF survey, and inventory and access results.

LLRP is an application layer protocol and does not provide retransmission, or reordering facilities. State consistency between the Client and the Reader is critical for the correct functioning of the system. Using LLRP messages, the Client updates the Reader state which includes Reader configuration parameters, dynamically created data structures (e.g., ROSpecs, AccessSpecs, etc), and possibly vendor defined data. For this reason, LLRP requires acknowledgements for the Client to Reader transactions – this provides a fail-safe mechanism at the LLRP layer to cope with network error situations. Also, to cope with intermittent connections, a Client can request a Reader's configuration state to confirm that a Reader's state is consistent with the Client after the Client reconnects (see LLRPConfigurationStateValue in section 12.2.1). The Reader to Client messages are primarily reports, status notifications or keepalives. Only the keepalives are acknowledged by the Client.

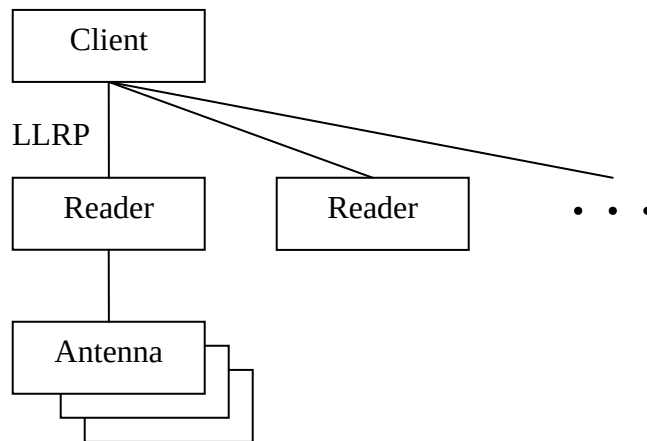


Figure 2: LLRP Endpoints

As shown in Figure 2, from LLRP's perspective, a Reader contains a collection of one or more antennas. Moreover, Readers as used in this specification may not necessarily be in one-to-one correspondence with hardware devices.

5.1 Typical LLRP Timeline

LLRP operation consists of the following phases of execution:

- Capability discovery
- Device configuration
- [optional] Inventory and access operations setup
- Inventory cycles executed
- If tag conditions matched, access operations will be executed during inventory cycle execution. Access operations include reading, writing, and locking tag memory, killing tags, etc.
- RF Survey operations executed
- Reports returned to the Client

A typical timeline of both LLRP and air protocol interactions between a Client, a Reader and a population of tags is depicted in Figure 3.

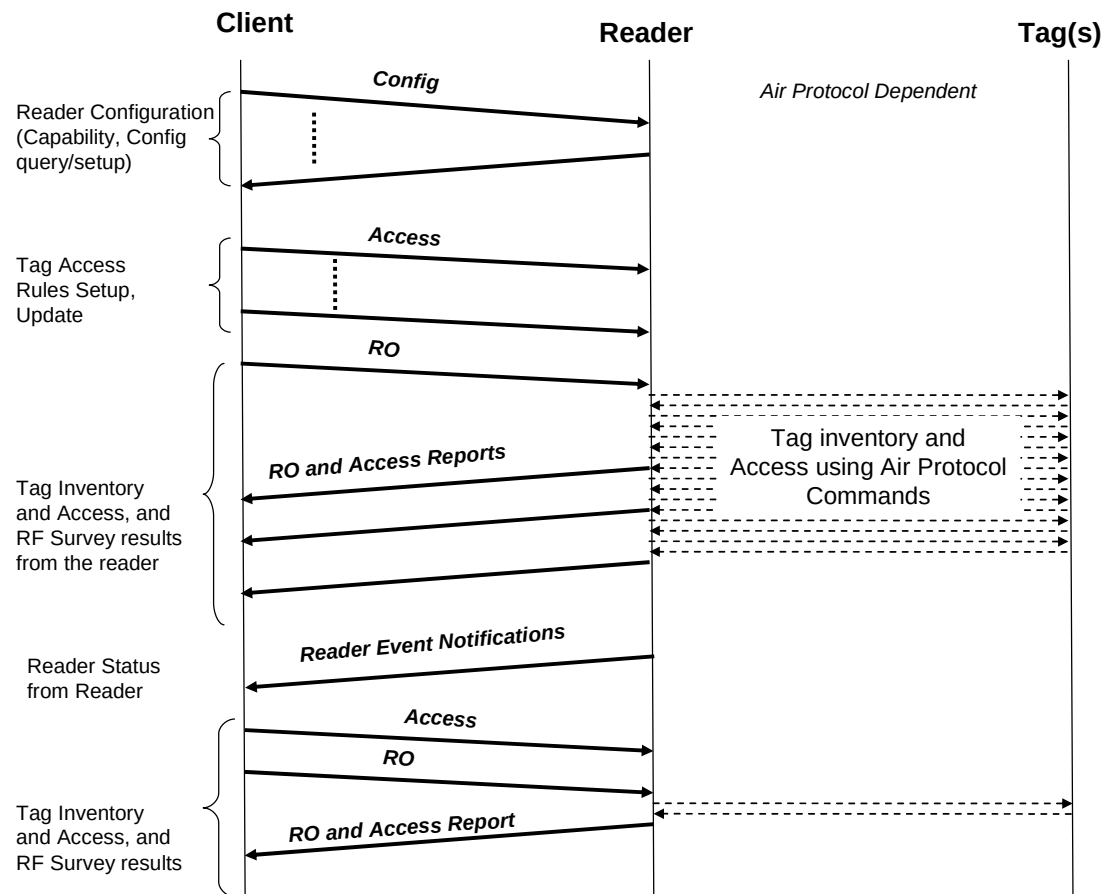


Figure 3: Typical LLRP Timeline

6 LLRP Operation

LLRP uses protocol data units called messages (See section 7.1.2 for details) to communicate between the Client and the Reader. Using LLRP, the Client updates or retrieves configuration of the Reader; by doing so, it controls the Reader's operation. This section provides an overview of the abstract model of the LLRP interface, and the data structures used in LLRP to and from the Reader.

Section 19 presents an informative description of the LLRP object model based upon UML notation.

6.1 Inventory, RF Survey and Access Operations

LLRP is based upon an abstraction of RFID air protocols and their respective commands. There are two principal concepts to the LLRP abstraction of RF operations by a Reader: 1) Reader Operations, and 2) Access operations. The remainder of this section provides a detailed description of these LLRP concepts.

Reader Operations (RO) define the parameters for operations such as Antenna Inventory and RF Survey. Access Operations define the parameters for performing data access operations to and from a tag.

The timing control of an operation is specified using boundary specification, which specifies how the beginning (using start trigger) and the end (using stop trigger) of the operation is to be determined.

An *antenna inventory* (AI) is the smallest unit of interaction between a Reader and tags in the antenna's *field-of-view* (FOV). An *InventoryParameterSpec* defines the parameters to be used during the inventory operation including protocol, protocol-specific parameters, and RF parameters. During an AI, the tags in the FOV of the antennas are singulated using air protocol commands based on the contents of the *InventoryParameterSpec*. An *AISpec* binds a stop trigger and a set of antennas to a set of *InventoryParameterSpecs*, and is identified by a Spec¹ identifier. The stop trigger defines the termination condition of the aggregate *AISpec* operation comprising of $N * M$ antenna inventory operations, where N and M are the cardinality of the antenna set and *InventoryParameterSpecs* set respectively. For example, if there is a single antenna and a single *InventoryParameterSpec* defined in an *AISpec*, the AI operation specified by the <antenna, *InventoryParameterSpec*> tuple is bounded by the stop trigger specification.

It should be noted that the stop trigger specification of each individual AI is not specified, which means the Reader is not limited to execute the AIs in the order in which they appear in an *AISpec*. The timing control and the sequencing of the individual AIs within an *AISpec* will be determined by the Reader.

RF Survey is an operation during which the Reader performs a scan and measures the power levels across a set of frequencies at an antenna. The RF survey operational parameters are described in a *RFSurveySpec* and it defines the survey operation at a single antenna. It comprises an identifier for the spec, an antenna identifier, stop trigger and set of parameters for the survey operation.

A *Reader Operation* (RO) describes the operations to be executed at one or more antennas of the Reader. A RO comprises at least one Spec, where a Spec is either an *AISpec* or a *RFSurveySpec*. If a RO comprises multiple Specs, each Spec is an *AISpec* or a *RFSurveySpec*. Each RO's operational parameters are described in a *ROSpec*. The *ROSpec* contains a spec identifier, the boundary specification for the entire RO operation, priority, a list of *AISpecs* and/or *RFSurveySpecs*, and optionally a reporting specification. The reporting specification defines the contents of RO Report and the trigger conditions when to send the inventory report and survey report. The order of *AISpec* and *RFSurveySpec* execution within a *ROSpec* is the order in which they appear in the *ROSpec*.

Figure 4 illustrates the statechart of a *ROSpec*. The *ROSpec* has three states: Disabled, Inactive and Active. The Client configures a new *ROSpec* using an ADD_ROSPEC message for the *ROSpec*. The *ROSpec* starts at the Disabled state waiting for the ENABLE_ROSPEC message for the *ROSpec* from the Client, upon which it transitions

¹ The document uses the term 'Spec' to denote the parameter specification for an operation.

to the Inactive state. The ROSpec does not respond to start or stop triggers in the Disabled state. The Client disables a *ROSpec* using a DISABLE_ROSPEC message for the *ROSpec*.

The *ROSpec* transitions from the Inactive state to the Active state when ROSpecStartCondition occurs for the *ROSpec*. The *ROSpec* transitions back to the inactive state when ROSpecDoneCondition happens.

ROSpecStartCondition = ROSpecStartTrigger or START_ROSPEC

ROSpecDoneCondition = AllSpecsDone or ROSpecStopTrigger or preempted or (STOP_ROSPEC message for the ROSpec from the Client)

The *ROSpec* when undefined is no longer considered for execution. The Client undefines the *ROSpec* using a DELETE_ROSPEC message for the *ROSpec*.

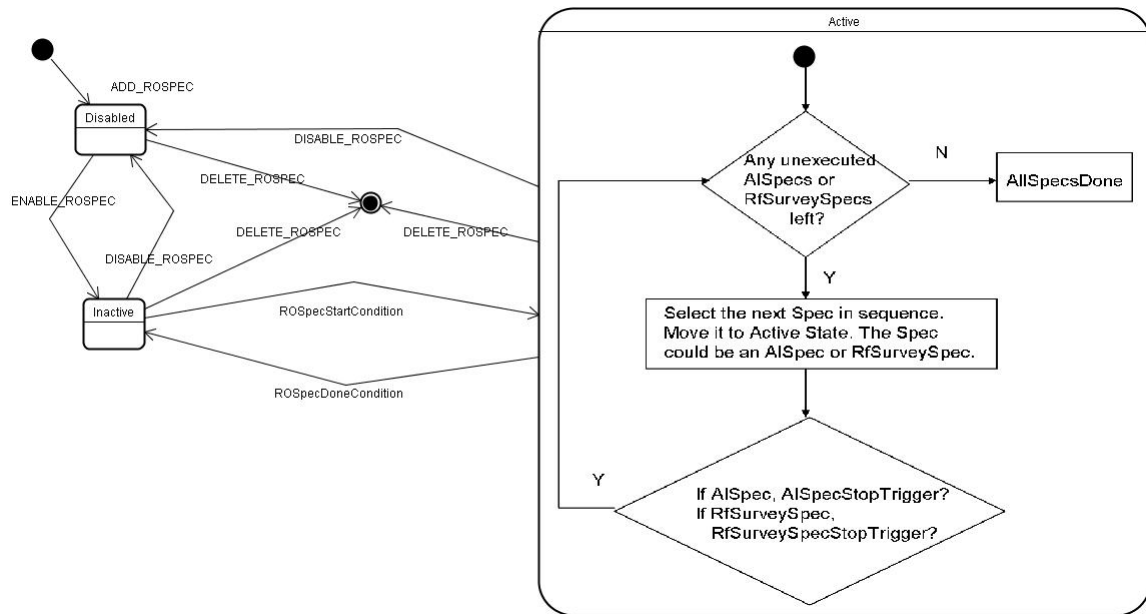


Figure 4: ROSpec Statechart

LLRP supports configuring multiple *ROSpecs*. Each *ROSpec* has a priority field. The default is for all the *ROSpecs* to have the same priority. Since the start trigger for the *ROSpec* can be an asynchronous event, there may be situations where a *ROSpec*'s start trigger event occurs when the Reader is busy executing another *ROSpec*. The Client, when setting up a *ROSpec*, can set the appropriate priority so that a high priority *ROSpec* can preempt a currently active lower priority *ROSpec* and start execution as soon as the ROSpecStartCondition for the higher priority, inactive *ROSpec* occurs. The *ROSpec* that got preempted transitions to the Inactive state.

Figure 5 illustrates the *AISpec* statechart. When the parent *ROSpec* moves to the active state, each *AISpec* in the *ROSpec* starts at the inactive state. During an active *ROSpec*'s

execution, when an inactive *AISpec* is selected for execution, that *AISpec* moves to the active state. If there are multiple antennas and *InventoryParameterSpecs* in that *AISpec*, the Reader picks the next $\langle \text{antenna}, \text{InventoryParameterSpec} \rangle$ to execute. In the figure, the ID of the selected antenna is A, and the protocol for the selected *InventoryParameterSpec* is P. The Reader starts tag singulation for air protocol P on antenna A using the operational parameters specified in the *InventoryParameterSpec*. This involves one or more air protocol commands from the Reader via the antenna to the tags in the antenna's FOV. The tags get singulated and each tag's EPC information is received by the antenna. If further tag memory operations are to be performed, such as writing or reading other memory regions, it will be performed at this point. As illustrated in Figure 6, these access operations are interleaved with the execution of an *AISpec*. Access operations are described using *AccessSpecs*. *AccessSpecs* describe the tags (*TagSpec*) on which some operations are to be performed, the operations to be performed (*OpSpec*), the boundary specification, and optionally a reporting specification for the Access operation. The *AccessSpec* may contain antenna information at which this access operation needs to be executed and contains the air protocol to be used to perform the access operations. In addition, to accommodate scenarios where an access operation needs to be performed only during a particular *ROSpec* execution, the *AccessSpec* optionally contains the *ROSpec* information. There can be one or more *AccessSpecs* set up at the Reader.

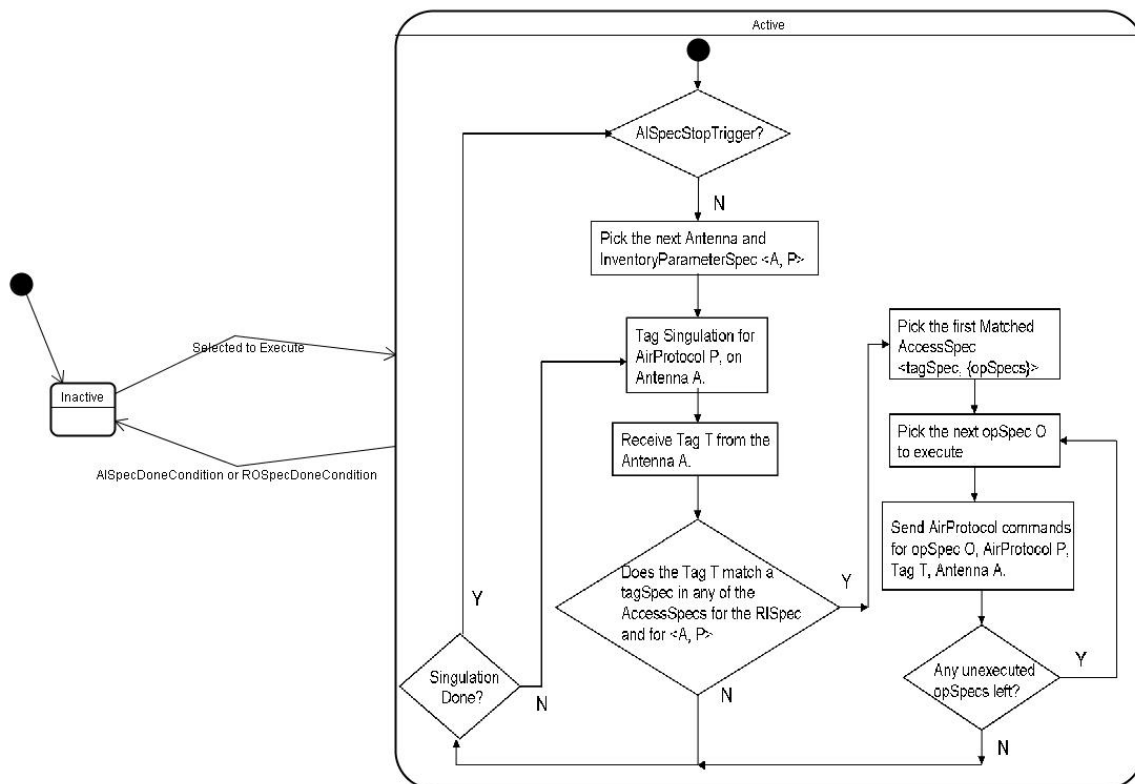


Figure 5: Antenna Inventory Spec States

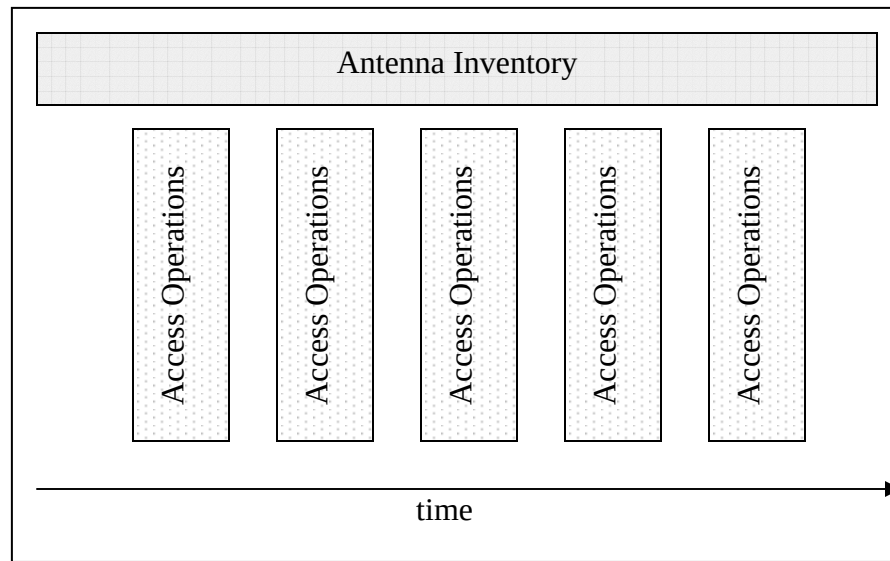


Figure 6: Access Operations Interleaved in an Antenna Inventory Operation

In Figure 5, when tags are received as a result of singulation, a check is performed to determine if the received tag matches the TagSpec defined in any of the Active (See statechart in Figure 8) *AccessSpecs*. In case there are multiple *AccessSpecs* that get matched during a tagSpec lookup, the Reader will execute the first *AccessSpec* that matches, where the ordering of the *AccessSpecs* is the order in which the *AccessSpecs* were created by the Client.

When an *AccessSpec* is executed, the set of operations as specified in *OpSpecs* of the *AccessSpec* are performed on the tag, which results in one or more air protocol commands and responses transacted between the Reader and the tag via antenna A over air protocol P. In order to support cases where the Reader needs to query the Client for further information to complete the operation on the tag, there is an *OpSpec* called the *ClientRequestOpSpec*.

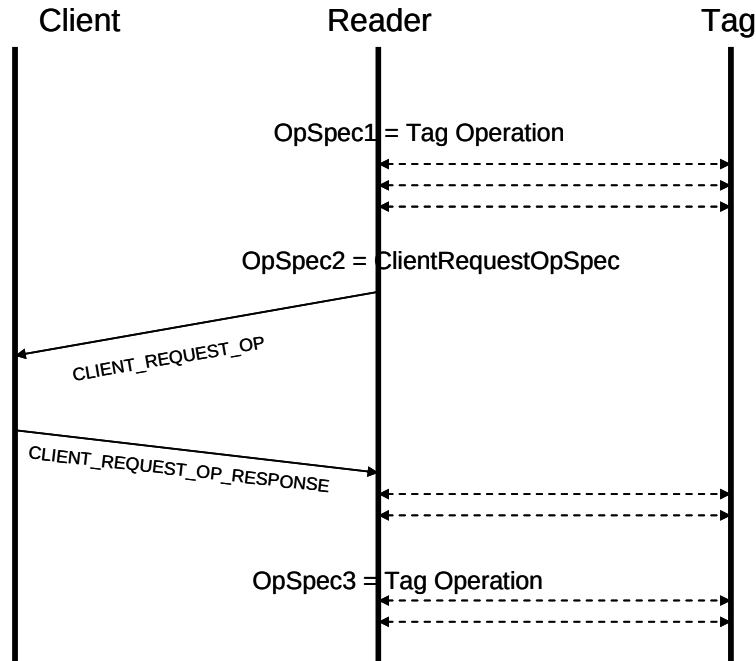


Figure 7: Client Request OpSpec

Figure 7 illustrates the message interaction between the Client, Reader and Tag for a ClientRequestOpSpec. For OpSpecs that are not ClientRequestOpSpec, the Reader performs the operations on the tag using the the air protocol commands. If an OpSpec is of the ClientRequestOpSpec, the Reader sends the result of the ongoing AccessSpec till that point in a CLIENT_REQUEST_OP message, so that the Client has all the relevant information to send a response. The client response is carried in a CLIENT_REQUEST_OP_RESPONSE message. This message is the set of OpSpecs that the reader should execute. The reader continues to execute the OpSpecs within an AccessSpec until all opSpecs have been executed or until an error occurs. When execution completes, the reader resumes the inventory operation.

The AISpec transitions back to the inactive state when AISpecDoneCondition occurs or when the parent ROspec's ROspecDoneCondition occurs.

AISpecDoneCondition = AISpecStopTrigger

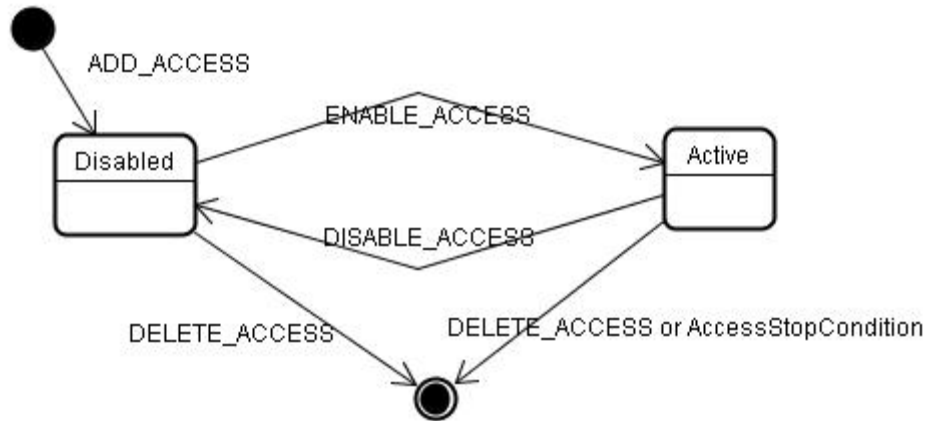


Figure 8: Access Spec States

Figure 8 illustrates the *AccessSpec*'s states. The Client configures an *AccessSpec* using an *ADD_ACCESS_SPEC* message for the *AccessSpec*. The *AccessSpec* starts at the Disabled state, waiting for an *ENABLE_ACCESS_SPEC* message from the Client for that *AccessSpec*, upon which it enters the Active state. It is only in the Active state that the *AccessSpec* is considered for execution. The Client can disable an *AccessSpec* using a *DISABLE_ACCESS_SPEC* message for the *AccessSpec*. The *AccessSpec* when undefined is no longer considered for execution. The Client undefines the *AccessSpec* using a *DELETE_ACCESS_SPEC* message for the *AccessSpec*.

In order for the Reader to take a local action to limit the validity of an *AccessSpec*, the Client can configure a stop trigger for the *AccessSpec*. An example use case of the stop trigger is when an *AccessSpec* is defined on all the antennas, and the desired behavior is to operate on the tag only once, the first time it is seen at any antenna. When the *AccessStopCondition* occurs, the *AccessSpec* transitions to undefined and is no longer considered for execution.

AccessStopCondition = AccessSpecStopTrigger

Figure 9 illustrates the *RFSurveySpec* statechart. When the parent *ROSpec* moves to the active state, each *RFSurveySpec* in the *ROSpec* starts at the inactive state. During an active *ROSpec*'s execution, when an inactive *RFSurveySpec* is selected for execution, that *RFSurveySpec* moves to the active state. In the active state, the Reader executes the survey operation as specified by the *RFSurveySpec*. The *RFSurveySpec* transitions back to the inactive state when the *RFSurveySpecDoneCondition* occurs or when the parent *ROSpec*'s *ROSpecDoneCondition* occurs.

RFSurveyStopCondition = RFSurveySpecStopTrigger

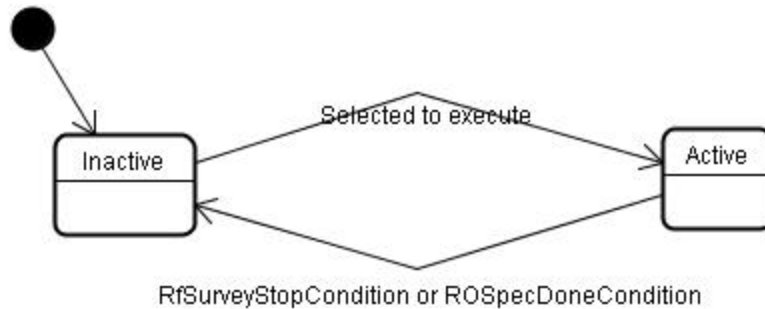


Figure 9: RFSurveySpec States

In summary, the Reader operation and Access operation specific data structures pass the following information between a Client and a Reader:

ROSpec: Details of a Reader operation

- ROSpecID: This identifier is generated by the Client. This identifier is used by the Client to perform operations on this ROSpec, like start, stop, enable, disable and delete. Reports that are generated as a result of the execution of this ROSpec also carry this identifier.
- ROBoundarySpec:
 - o ROSpecStartTrigger, ROSpecStopTrigger: This is the start and stop trigger for this ROSpec. The triggers that are specifiable for a ROSpec are listed in Table 1.
- Priority: This is the priority of this ROSpec.
- CurrentState: This is the current state of the ROSpec – disabled, inactive, active. This field is kept up to date by the Reader based on the ROSpec’s current state.
- Set of Specs: Each Spec is either an *AISpec* or *RFSurveySpec*. The Specs are executed in the order in which it is defined in the ROSpec. The position of the Spec (*AISpec* or *RFSurveySpec*) in the ROSpec is called the SpecIndex. The SpecIndex is used during reporting to identify the spec inside of ROSpec whose execution generated the data in the report. The numbering of SpecIndex is 1 based.
- ROReportingSpec: If specified, this defines when to send the results of this ROSpec, and also the contents and format of the report.

AISpec: Details of one of more antenna inventory operations

- AISpecStopTrigger: This is the stop trigger for the AISpec. The triggers that are specifiable for an AISpec are listed in Table 1.
- Set of Antenna IDs: This is the set of antennas at which the inventory operations described in the InventoryParameterSpecs are executed. If there are N antennas and M InventoryParameterSpecs, the Reader will execute the M inventory operations at each of the specified antennas. Thus, in aggregate, the Reader will

875 execute N * M AIs (Antenna inventory operations). The ordering of the AIs is
876 determined by the Reader.

- 877 • Set of InventoryParameterSpecs: There can be one or more
878 InventoryParameterSpecs specified as part of the AISpec. Collectively, they are
879 bound by the AISpecStopTrigger. The order in which the antenna inventory
880 operations described as <Antenna, InventoryParameterSpec> are executed is
881 determined in a proprietary manner inside the Reader.

882

883 **InventoryParameterSpec:** Operational parameters for an inventory using a single air
884 protocol.

- 885 • InventoryParameterSpecID: This identifier is generated by the Client. Reports that
886 are generated as a result of the execution of this InventoryParameterSpec carry
887 this identifier.

- 888 • Air Protocol: This is the air protocol that is used to inventory the tags in the field
889 of view of the antenna.

- 890 • Set of Antenna Configuration Settings: Each Antenna Configuration setting
891 comprises of

- 892 o Antenna ID: The identifier of the antenna

- 893 o RFTransmitterSettings: This describes the configuration of the transmitter
894 during the inventory operation.

- 895 o RFReceiverSettings: This describes the configuration of the receiver
896 during the inventory operation.

- 897 o AirProtocolInventoryCommandSettings parameters: This describes the
898 configuration of the air protocol parameters for the inventory operation.

899

900 **RFSurveySpec:** Details of a RF Survey operation

- 901 • RFSurveySpecID: This identifier is generated by the Client. Reports that are
902 generated as a result of the survey operation carry this identifier.

- 903 • RFSurveySpecStopTrigger: This is the stop trigger for the RFSurveySpec. The
904 triggers that are specifiable for a RFSurveySpec are listed in Table 1.

- 905 • AntennaID: This is the antenna at which the survey operation is to be executed.

- 906 • StartFrequency: This is the starting channel for which power levels need to be
907 measured during this RF survey operation.

- 908 • EndFrequency: This is the ending channel for which power levels need to be
909 measured during this RF survey operation. The RF survey operation is performed
910 on frequency channels between the specified Start Frequency and End frequency.

911

912 **AccessSpec:** Details of an access operation.

- 913 • AccessSpecID: This identifier is generated by the Client upon creation of this
914 AccessSpec. This identifier is used by the Client to perform operations on this
915 AccessSpec, like start, stop and delete. Reports that are generated as a result of
916 the execution of this AccessSpec also carry this identifier.
- 917 • AntennaID: This is the identifier of the antenna for whose tag observations this
918 AccessSpec is executed.
- 919 • Air Protocol: This is the air protocol used to perform access operations on the tag.
- 920 • ROSpecID: This is the identifier of the ROSpec during whose tag observations
921 this AccessSpec is executed.
- 922 • CurrentState: This is the current state of the AccessSpec – disabled, active. This
923 field is kept up to date by the Reader based on the AccessSpec's current state.
- 924 • AccessSpecStopTrigger: If specified, this is the trigger to undefine the
925 AccessSpec upon the occurrence of the stop trigger.
- 926 • AccessCommand: This parameter is used to configure the air protocol parameters
927 for the access operation. At a minimum, this specifies the tag filters for which the
928 access operations are to be performed, and the list of operations to be performed
929 on the tag.
 - 930 o TagSpec: This describes the tag filters and is specified in terms of the air
931 protocol's tag memory layout.
 - 932 o List of OpSpecs: This is specified in terms of the air protocol's tag access
933 operations. The order of execution is determined by the order in which it is
934 configured in the AccessSpec.
- 935 • AccessReportSpec: If specified, this defines when to send the results of this
936 AccessSpec, and also the contents and format of the report.

937 6.1.1 Operation Triggers

938 This section describes the triggers that can be configured using LLRP to control the
939 various operations.

940 6.1.1.1 Summary

941 The specific triggers used to control the various operations are presented in a tabular
942 fashion.

943 **Table 1: Operation Triggers**

<i>Trigger Name</i>	<i>ROSpecStart</i>	<i>ROSpecStop</i>	<i>AISSpecStop</i>	<i>AccessSpecStop</i>	<i>RFSurveySpec Stop</i>
GPI Trigger	X	-	-	-	-
GPI Trigger with Timeout	-	X	X	-	-
N attempts	-	-	X	-	-

N tag observations	-	-	X	-	-
No tag observations for t ms	-	-	X	-	-
Immediate	X	-	-	-	-
Null	X	X	X	X	X
Time Based Periodic	X	-	-	-	-
Time Based Duration	-	X	X	-	X
Operation Count	-	-	-	X	X

944

945 **6.1.1.2 Reader Operation Triggers**

946 The triggers SHALL operate as follows:

- 947 • Null: When used as a start or a stop trigger, it implies no start or stop conditions
948 have been specified, respectively.
- 949 • Immediate: This is used as a start trigger. Operations using this trigger will start
950 immediately.
- 951 • Time-based: There are two different types of time-based triggers defined in LLRP
952 – periodic and duration.
 - 953 o Periodic: This is used as a start trigger. This is specified using UTC time
954 [UTC], offset and period. For one-shot inventory, period is set to 0, and
955 for periodic inventory operation, period > 0. If UTC time is not specified,
956 the first start time is determined as (time of message receipt + offset), else,
957 the first start time is determined as (UTC time + offset). Subsequent start
958 times = first start time + k * period (where, k > 0).
 - 959 o Duration: This is used as a stop trigger.
- 960 • Tag observation based: There are three different types of tag-observation based
961 triggers defined in LLRP. They are all used only as stop triggers. Each of these
962 trigger types have a timeout value. So the trigger event happens when either the
963 tag observation event happens or the timeout expires.
 - 964 o Upon seeing N tags, or timeout.
 - 965 o Upon seeing no more new tags for t milliseconds, or timeout
 - 966 o N attempts to see all the tags in the field of view, or timeout
- 967 • External events: These are due to events received at Reader interfaces like signal
968 transition on a GPI port or a message on the network port.
 - 969 o GPI event at a GPI port, or a timeout
 - 970 o Client triggers: A Client can instruct the Reader to start/stop a particular
971 operation using LLRP messages.

- Operation count: This is used as a stop trigger for RFSurvey. This trigger limits the number of times the Reader takes survey measurements across the specified frequency range.

AI and RFSurvey specs do not contain start triggers. The first spec (AISpec or RFSurveySpec) starts when the ROSpec enters the active state. The kth Spec in the ROSpec starts immediately after the completion of the k-1th Spec.

When Null is specified as a stop trigger for a Spec ((either AISpec or RFSurveySpec), the execution of the Spec is stopped only when the parent ROSpec's ROSpecDoneCondition occurs.

6.1.1.3 Access Operation Triggers

AccessSpecs do not contain start triggers. An AccessSpec when enabled using ENABLE_ACCESS_SPEC will transition to the active state. There is only one type of stop trigger for controlling the validity of an AccessSpec:

- Operation count: This is used as a stop trigger. This trigger is useful to limit the number of times the instance of the operation is executed during its lifetime.

6.2 Reporting, Event Notification and Keepalives

The results of the inventory, access and RF survey operations, will be sent by the Reader to the Client in the form of reports. Using LLRP, the Client is capable of setting up the triggers that determine when the report is to be sent by the Reader, and also the contents and format of the report. The report message is RO_ACCESS_REPORT. The triggers and report contents can be configured in one of the following ways:

- Differently for each ROSpec and AccessSpec when creating them using the ADD_ROSPEC and ADD_ACCESSSPEC messages, respectively.
- Global default using the SET_READER_CONFIG message.

Table 2 summarizes the triggers available in LLRP to control when the RO report and the AccessReport is to be generated and sent by the Reader.

Table 2: Reporting Triggers

Trigger Name	ROReport	AccessReport
None	X	-
(Upon N tags or End of Spec), where Spec = AISpec or RFSurveySpec	X	-
Upon N tags or End of ROSpec	X	-
End of AccessSpec	-	X
Whenever ROReport is generated for the RO that triggered the execution of	-	X

this AccessSpec		
-----------------	--	--

999

1000 In addition to data reports, the Client can configure the Reader to enable or disable
 1001 notification of events as and when it happens at the Reader. Some examples of events are
 1002 frequency hop, buffer overflow, etc.

1003 In order to monitor the LLRP-layer connectivity with the Reader, the Client can
 1004 configure the Reader to send Keepalives periodically. The Keepalive message is
 1005 acknowledged by the Client, using which, the Reader can also monitor the LLRP-layer
 1006 connectivity with a Client. The Keepalives can be disabled. If enabled, the periodicity of
 1007 the message is specified by the Client.

1008 **7 Messages, Parameters and Fields**

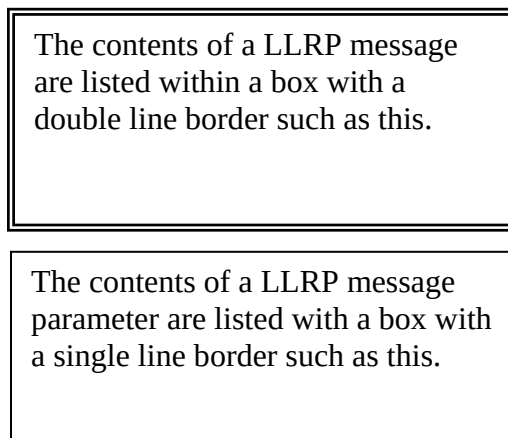
1009 LLRP is a message-oriented protocol made up of data elements called protocol data units.
 1010 This section provides the details of each message type and parameter type, and expresses
 1011 them in an abstract manner. The section starts with an overview of the message types and
 1012 parameters, where the messages are grouped into separate functional groups.

1013 **7.1 Overview**

1014 LLRP provides an extensible mechanism to support existing and new air protocols. It is
 1015 achieved by decoupling messages from parameters – using a common message structure
 1016 across air protocols, and providing extensibility in the form of parameters.

1017 **7.1.1 Formatting Conventions**

1018 LLRP messages and parameters are defined using the graphical notation below.



1019

1020 **Figure 10: Box Formats for Messages and Parameters**

1021

1022 Contained within the box is an ordered list of sub-parameters and fields contained within
 1023 the message or parameter. The field/parameter names are shown in **boldface**, followed

1024 by the data type and a brief description of the field/parameter when necessary. Fields
1025 with values that are restricted to a subset of the range of their data type have their
1026 possible and legal values shown in *italics* below the field name.

1027

1028 Fields are composed of one of the following basic data types:

1029 **Bit** – An integer with only two possible values, 0 or 1

1030 **Bit Array** – A sequence of bits.

1031 **Byte Array** – A sequence of bytes.

1032 **Boolean** – A field that can take the values TRUE or FALSE.

1033 **Integer** – An integer can take any whole number. When this value is used in the abstract
1034 specification, the *Possible Values* element will specify the possible and legal value for a
1035 particular field.

1036 **Short Array** – A sequence of unsigned short integers

1037 **Signed Integer** – A signed integer can take any whole number value between -2^{31} through
1038 $2^{31}-1$ inclusive. Within the abstract specification, the *Possible Values* element will
1039 enumerate any restrictions beyond these limits for a particular field.

1040 **Signed Short Integer** – A signed short integer can take any whole number value between
1041 -2^{15} through $2^{15}-1$ inclusive. Within the abstract specification, the *Possible Values*
1042 element will enumerate any restrictions beyond these limits for a particular field

1043 **Unsigned Integer** – An unsigned integer is a value that is between 0 through $2^{32}-1$
1044 inclusive. Within the abstract specification, the *Possible Values* element will enumerate
1045 any restrictions beyond these limits for a particular field.

1046 **Unsigned Long Integer** – An unsigned long integer is a value that is between 0 through
1047 $2^{64}-1$ inclusive. Within the abstract specification, the *Possible Values* element will
1048 enumerate any restrictions beyond these limits for a particular field.

1049 **Unsigned Short Integer** – An unsigned short integer is a value that is between 0 through
1050 $2^{16}-1$ inclusive. Within the abstract specification, the *Possible Values* element will
1051 enumerate any restrictions beyond these limits for a particular field.

1052 **UTF-8 String** – A sequence of UTF-8 [UTF8] encoded characters.

1053 In addition to the basic types, fields can be defined as ‘lists’ of a basic type. A list is an
1054 ordered set of a basic type. The order is preserved by all bindings.

1055 **7.1.2 Messages**

1056 Each Message contains:

- 1057 • Version value that indicates the version of the protocol for this message.
- 1058 • Message Type value that uniquely identifies it within a protocol message.
- 1059 • Message ID: The Reader behavior SHALL be based upon starting the processing
1060 of messages in the order received over LLRP, however, the completion of

1061 execution of the message processing MAY not necessarily be in the same order
1062 inside the Reader. Hence, the Reader responses to the messages may be in a
1063 different order than the order of the Client messages. The Message ID is to
1064 facilitate multiple outstanding messages/requests from Client or Reader. The
1065 communications between the Client and the Reader is primarily of a request-
1066 response type - requests/commands from the Client to the Reader, and response
1067 from the Reader to the Client. The Message ID is used to associate a Reader
1068 response with the original Client message.

- 1069 • In addition, it may contain mandatory or optional parameters.

1070 **7.1.3 Parameters**

1071 LLRP Parameters are used to communicate specific details of LLRP operation in LLRP
1072 Messages. Each Parameter contains:

- 1073 • Parameter Type value that uniquely identifies it within a Message.
- 1074 • In addition, it may contain individual fields or sub-parameters.

1075 **7.1.3.1 General Parameters**

1076 This section describes the set of parameters that are used in multiple messages or
1077 parameters.

1078 **7.1.3.1.1 Timestamp**

1079 The timestamps in LLRP messages or parameters can be either the uptime or the UTC
1080 time [UTC]. If a Reader has an UTC clock, all timestamps reported by the Reader
1081 SHALL use an UTC timestamp parameter. If a Reader has no UTC clock capability, all
1082 timestamps reported by the Reader SHALL use the uptime parameter.

1083 **7.1.3.1.1.1 UTCTimestamp Parameter**

1084 **Compliance requirement:** Compliant Readers and Clients that have UTC clocks
1085 SHALL implement this parameter.

UTCTimestamp Parameter

MicroSeconds: Unsigned Long Integer. This is the time elapsed since the Epoch
(00:00:00 UTC, January 1, 1970) measured in microseconds.

1086 **7.1.3.1.1.2 Uptime Parameter**

1087 **Compliance requirement:** Compliant Readers and Clients that do not have UTC clocks
1088 SHALL implement this parameter. Compliant Readers and Clients that have UTC clocks
1089 MAY implement this parameter.

Uptime Parameter

Microseconds: Unsigned Long Integer. This is the time elapsed since boot, measured in microseconds.

7.1.4 Fields

Messages and parameters may contain individual fields. In this section, we present the enumerations and the interpretation of the value of zero for certain fields.

ProtocolID: This is the identifier of the air protocol. The air protocol enumerations used in the LLRP protocol are as follows:

Table 3: Air Protocol Enumerations used in LLRP

Air Protocol Enumerations

Protocol ID : Integer

Possible Values:

Value	Protocol
-----	-----
0	Unspecified air protocol
1	EPCGlobal Class 1 Gen 2
2-255	Reserved for future use

Compliance requirement: Compliant Readers and Clients SHALL use this enumeration.

AntennaID, ROSpecID, AccessSpecID, GPIPort, GPOPort: These fields are identifiers for LLRP-related objects within the Reader. For example, AntennaID is the identifier of the antenna; ROSpecID is the identifier of the ROSpec. The objects are indexed from 1. A value of non-zero for a field is a specific instance of the respective object. A value of zero means all instances of the respective object.

7.1.5 Functional Grouping

The LLRP messages are grouped into:

- **Reader device capabilities:** Messages that query Reader capabilities. They include
 - o GET_READER_CAPABILITIES
 - o GET_READER_CAPABILITIES_RESPONSE
- **Reader operations control:** Messages that control the Reader's air protocol inventory and RF operations. They include
 - o ADD_ROSPEC
 - o ADD_ROSPEC_RESPONSE
 - o DELETE_ROSPEC

- 1114 o DELETE_ROSPEC_RESPONSE
- 1115 o START_ROSPEC
- 1116 o START_ROSPEC_RESPONSE
- 1117 o STOP_ROSPEC
- 1118 o STOP_ROSPEC_RESPONSE
- 1119 o ENABLE_ROSPEC
- 1120 o ENABLE_ROSPEC_RESPONSE
- 1121 o DISABLE_ROSPEC
- 1122 o DISABLE_ROSPEC_RESPONSE
- 1123 o GET_ROSPECS
- 1124 o GET_ROSPECS_RESPONSE
- 1125 • **Access control:** Messages that control the tag access operations performed by the
- 1126 Reader. They include
 - 1127 o ADD_ACCESSSPEC
 - 1128 o ADD_ACCESSSPEC_RESPONSE
 - 1129 o DELETE_ACCESSSPEC
 - 1130 o DELETE_ACCESSSPEC_RESPONSE
 - 1131 o ENABLE_ACCESSSPEC
 - 1132 o ENABLE_ACCESSSPEC_RESPONSE
 - 1133 o DISABLE_ACCESSSPEC
 - 1134 o DISABLE_ACCESSSPEC_RESPONSE
 - 1135 o GET_ACCESSSPECS
 - 1136 o GET_ACCESSSPECS_RESPONSE
 - 1137 o CLIENT_REQUEST_OP
 - 1138 o CLIENT_REQUEST_OP_RESPONSE
- 1139 • **Reader device configuration:** Messages that query/set Reader configuration, and
- 1140 close LLRP connection. They include
 - 1141 o GET_READER_CONFIG
 - 1142 o GET_READER_CONFIG_RESPONSE
 - 1143 o SET_READER_CONFIG
 - 1144 o SET_READER_CONFIG_RESPONSE
 - 1145 o CLOSE_CONNECTION
 - 1146 o CLOSE_CONNECTION_RESPONSE

- 1147 • **Reports:** These are messages that carry different reports from the Reader to the
1148 Client. Reports include Reader device status, tag data, RF analysis report. They
1149 include
 - 1150 o GET_REPORT
 - 1151 o RO_ACCESS_REPORT
 - 1152 o READER_EVENT_NOTIFICATION
 - 1153 o KEEPALIVE
 - 1154 o KEEPALIVE_ACK
 - 1155 o ENABLE_EVENTS_AND_REPORTS
 - 1156 • **Custom Extension:** This is a common mechanism for messages that contain
1157 vendor defined content.
 - 1158 o CUSTOM_MESSAGE
 - 1159 • **Errors:** Typically the errors in the LLRP defined messages are conveyed inside
1160 of the responses from the Reader. However, in cases where the message received
1161 by the Reader contains an unsupported message type, or a CUSTOM_MESSAGE
1162 with unsupported parameters or fields, the Reader SHALL respond with this
1163 generic error message.
 - 1164 o ERROR_MESSAGE
- 1165 LLRP parameters are used to communicate specific settings of LLRP operation in the
1166 messages. A parameter contains one or more fields, and in some cases also may nest one
1167 or more other parameters.
- 1168 Typically, each message type has its own set of parameters; however, there may be
1169 exceptions in some cases, where two different message types use the same parameter
1170 because they require the same setting exposed by the parameter.

1171 7.1.6 LLRP Messages and Actions

1172 This section describes the corresponding LLRP-related actions in the Reader upon
1173 receiving the various LLRP protocol messages. Figure 11 uses UML synchronous
1174 messaging notation. Messages are asynchronous.

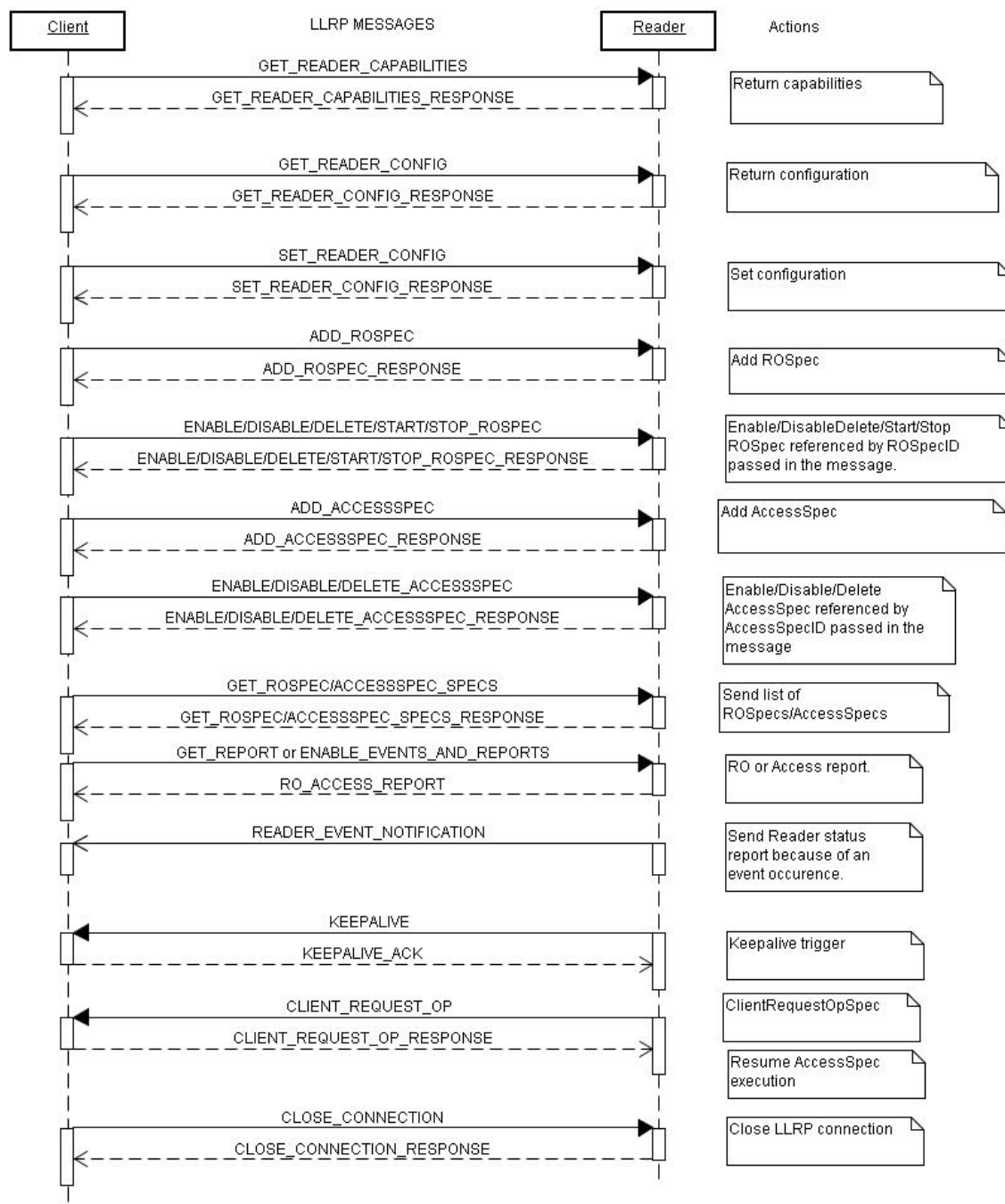


Figure 11: LLRP Messages and Reader Actions

8 Custom Extension

LLRP supports vendor extensions for defining commands and parameters within certain commands. All LLRP bindings support these extension mechanisms.

8.1 CUSTOM_MESSAGE

This message carries a vendor defined format from Reader to Client or Client to Reader. In addition to the version and messageID, the custom message also carries the information below.

CUSTOM_MESSAGE

Vendor Identifier: Unsigned Integer. IANA Private Enterprise Number

Message Subtype: Integer

Possible Values: 0-255.

Data: vendor specific format

No requirements are made as to the content or parameters contained within the Data portion of these messages. Clients MAY ignore CUSTOM_MESSAGES. Readers SHALL accept CUSTOM_MESSAGE and return an ERROR_MESSAGE if CUSTOM_MESSAGE is unsupported by the Reader or the CUSTOM_MESSAGE contains fields and/or parameters that are unsupported by the Reader.

8.2 Custom Parameter

Certain Messages and Parameter Sets within LLRP allow for the insertion of vendor defined parameters. These custom parameters have the following format.

Custom Parameter

Vendor Identifier: Unsigned Integer. IANA Private Enterprise Number

Parameter Subtype: Integer

Possible Values: 0-255

Data: vendor specific format

Clients SHALL accept messages (except for CUSTOM_MESSAGE) that contain custom parameters but MAY ignore all custom parameters within these messages. Readers SHALL accept messages (except for CUSTOM_MESSAGE) that contain custom parameters and SHALL return an error when such parameters are unsupported.

8.3 Custom Extension in Commands

The following commands allow one or more custom Parameters in their message structure:

GET_READER_CAPABILITIES

GET_READER_CONFIG

1203 GET_READER_CAPABILITIES_RESPONSE
1204 GET_READER_CONFIG_RESPONSE
1205 SET_READER_CONFIG

1206 **8.4 Custom Extension in Individual LLRP Parameters**

1207 LLRP only allows extension to parameters where the parameter set is defined with a
1208 custom Parameter type in the abstract model. All custom extension points will be marked
1209 in the abstract standard using the notation

1210

1211 **Custom Extension Point List:** List of <custom Parameter> [optional]

1212 The following example illustrates a fictitious parameter that allows the embedding of
1213 custom extension parameters.

Example Parameter

Field1: Unsigned Integer

relatedData: Example Sub Parameter

Custom Extension Point List: List of <custom Parameter> [optional]

1214

1215 This example shows that the Example Parameter could contain an optional custom
1216 parameter that must adhere to the custom Parameter format.

1217 **8.5 Allowable Parameter Extension**

1218 All parameter values are specified within the abstract binding. A Reader or Client
1219 SHALL NOT extend the range of fields defined within the abstract specification unless
1220 the possible values indicate ranges for user defined options.

1221 For example, the Identification Parameter defines a field to carry the ID type.

1222 **IDType:** Integer

1223 *Possible Values:*

1224	IDType	ID
1225	-----	--
1226	0	MAC address
1227	1	EPC

1228

1229 A Client or Reader adhering to the standard SHALL generate an **IDType** field with only
1230 those values shown (0-1). A Reader or Client implementation SHALL generate an error
1231 upon receiving a value outside this range.

9 Reader Device Capabilities

There are four broad categories of capabilities that are advertised by the Reader: general device, LLRP, regulatory, and air protocol capabilities.

9.1 Messages

9.1.1 GET_READER_CAPABILITIES

This message is sent from the Client to the Reader. The Client is able to request only a subset or all the capabilities from the Reader.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

GET_READER_CAPABILITIES

RequestedData: Integer

Possible Values:

Value	Requested Data
-----	-----
0	All
1	General Device Capabilities
2	LLRP Capabilities
3	Regulatory Capabilities
4	Air Protocol LLRP Capabilities

Custom Extension Point List: List of <custom Parameter> [optional]

9.1.2 GET_READER_CAPABILITIES_RESPONSE

This is the response from the Reader to the GET_READER_CAPABILITIES message. The response contains the LLRPStatus Parameter and the list of parameters for the requested capabilities conveyed via RequestedData in the GET_READER_CAPABILITIES message.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

GET_READER_CAPABILITIES_RESPONSE

Response Data: Set of LLRP Parameters.

Possible Values: The possible members are

<LLRPStatus Parameter>, and,

one or more from the set

< GeneralDeviceCapabilities Parameter,

LLRPCapabilities Parameter,

RegulatoryCapabilities Parameter,

AirProtocolLLRPCapabilities Parameter >.

Custom Extension Point List: List of <custom Parameter> [optional]

1249 9.2 Parameters

1250 9.2.1 GeneralDeviceCapabilities Parameter

1251 This parameter carries the general capabilities of the device like supported air protocols,
1252 version of the Reader firmware, device hardware and software information, and receive
1253 sensitivity table.

1254 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1255 parameter.

GeneralDeviceCapabilities Parameter

Device manufacturer name: Unsigned Integer. The IANA Private Enterprise Number (PEN).

Model name: Unsigned Integer

Firmware version: UTF-8 String

Maximum number of antennas supported: Unsigned Short Integer

CanSetAntennaProperties: Boolean. If set to true, the Client can set antenna properties (Section 12.2.5), else, the Client can not set it, but only query it using GET_READER_CONFIG.

Receive Sensitivity Table: List of <ReceiveSensitivityTableEntry Parameter>

Per Antenna Receive Sensitivity Range: List of <PerAntennaReceiveSensitivityRange Parameter>

Air protocol supported per antenna: N instances of <PerAntennaAirProtocol Parameter>, where N = Maximum number of antennae supported.

GPIO Support: <GPIO Capabilities Parameter>

HasUTCClockCapability: Boolean. If set to true, the Reader reports time based on

UTC timestamps (Section 7.1.3.1.1.1) in its reports, else, the Reader reports time based on Uptime (Section 7.1.3.1.1.2) in its reports.

9.2.1.1 ReceiveSensitivityTableEntry Parameter

This parameter specifies the index into the Receive Sensitivity Table for a receive sensitivity value. The receive sensitivity is expressed in dB and the value is relative to the maximum sensitivity. If the Reader does not allow control of receive sensitivity, a table of one entry is returned, the entry having the value of zero.

If the Reader allows control of receive sensitivity and the Reader also supports multiple antennas where the antennas can have different receive sensitivity values, then the Receive Sensitivity Table should be a set of values representing the union of sensitivity values for all antennas.

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

ReceiveSensitivityTableEntry Parameter

Index: Unsigned Short Integer

Receive sensitivity value: Integer. The value is in dB relative to the maximum sensitivity.

Possible Values: -128 to 0.

9.2.1.2 PerAntennaReceiveSensitivityRange Parameter

For a particular antenna, this parameter specifies the Reader's valid index range in the Receive Sensitivity Table. A Reader should report this parameter if the Reader allows control of receive sensitivity (i.e., the Reader reports a Receive Sensitivity Table with more than one entry) and the Reader supports multiple antennas where the antennas can have different receive sensitivity values.

If this parameter is omitted, then the Client SHALL assume that for all of the Reader's antennas the index range is the same as in the Receive Sensitivity Table.

Compliance requirement: Compliant Readers and Clients MAY implement this parameter.

PerAntennaReceiveSensitivityRange Parameter

Antenna ID: Unsigned Short Integer

Possible Values:

1 to N, where N is the maximum number of antennas supported by the device.

ReceiveSensitivityIndexMin: Unsigned Short Integer

Possible Values:

0 to S, where S is the number of Receive Sensitivity Table entries reported by the

Reader.

ReceiveSensitivityIndexMax: Unsigned Short Integer

Possible Values:

Mn to S, where Mn is the ReceiveSensitivityIndexMin and S is the number of Receive Sensitivity Table entries reported by the Reader.

1277 **9.2.1.3 PerAntennaAirProtocol Parameter**

1278 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1279 parameter.

PerAntennaAirProtocol Parameter

Antenna ID: Unsigned Short Integer

Possible Values:

1 to N, where N is the maximum number of antennas supported by the device.

Air protocols supported: List of Protocol Ids enumerated based on Table 3.

1280 **9.2.1.4 GPIOCapabilities Parameter**

1281 This parameter describes the GPIO capabilities of the Reader. A value of zero for
1282 NumGPIOs indicates that the Reader does not have general purpose inputs. A value of zero
1283 for NumGPOs indicates that the Reader does not have general purpose outputs.

1284
1285 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1286 parameter.

GPIOCapabilities Parameter

NumGPIOs: Unsigned Short Integer. Number of general purpose inputs supported by the device.

NumGPOs: Unsigned Short Integer. Number of general purpose outputs supported by the device.

1287 **9.2.2 LLRPCapabilities Parameter**

1288 This parameter describes the LLRP protocol capabilities of the Reader. These include
1289 optional LLRP commands and parameters, capacities of data structures used in LLRP
1290 operations, and air protocol specific capabilities used by LLRP.

1291 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1292 parameter. Readers MAY support RFSurvey, MAY support tag inventory state aware
1293 singulation, MAY support UTC clocks, MAY support buffer fill warning reports, MAY
1294 support EventAndReportHolding upon reconnect, and MAY support
1295 ClientRequestOpspec. Readers SHALL support at least one ROSpec, one AISpec per
1296 ROSpec, one InventoryParameterSpec per AISpec, one AccessSpec, and one OpSpec per
1297 AccessSpec.

LLRPCapabilities Parameter

CanDoRFSurvey: Boolean. If set to true, the Reader can perform RFSurvey operations (Section 10.2.3).

CanDoTagInventoryStateAwareSingulation: Boolean. If set to true, the Reader can support tag inventory state aware singulation.

CanReportBufferFillWarning: Boolean. If set to true, the Reader can report buffer fill warning in the reader event notification (Section 13.2.6.5).

MaxNumROSpecs: Integer. If zero, there is no limit. This is the maximum number of ROSpecs that can be configured at the Reader.

MaxNumSpecsPerROSpec: Integer. If zero, there is no limit. This is the maximum number of Specs (either AISpec or RFSurveySpec) that can be configured as part of a ROSpec at the Reader.

MaxNumInventoryParameterSpecsPerAISpec: Integer. If zero, there is no limit. This is the maximum number of InventoryParameterSpecs that can be configured per AISpec.

MaxPriorityLevelSupported: Integer. This is the maximum priority level supported in the reader. If set to less than or equal to 1, the Reader has no preemption support.

Possible Values: 0-7.

MaxNumAccessSpecs: Integer. If zero, there is no limit. This is the maximum number of AccessSpecs that can be configured at the Reader.

MaxNumOpSpecsPerAccessSpec: Integer. If zero, there is no limit. This is the maximum number of OpSpecs that can be configured per AccessSpec at the Reader.

SupportsClientRequestOpSpec: Boolean. If set to true, the Reader supports client request OpSpecs (Section 11.2.1.2.1).

ClientRequestOpSpecTimeout: Unsigned Short Integer (in milliseconds). The time the Reader will wait for the CLIENT_REQUEST_OP_RESPONSE from the Client after sending a RO_ACCESS_REPORT message upon executing the ClientRequestOpSpec OpSpec. This field is valid only if the Reader supports ClientRequestOpSpec (Section 11.2.1.2.1). If this field is 0, there is no limit.

SupportsEventAndReportHolding: Boolean. If set to True, the Reader supports the EventsAndReports Parameter and the ENABLE_EVENTS_AND_REPORTS message. If set to false, the Reader does not support the ENABLE_EVENTS_AND_REPORTS message or the EventsAndReports Parameter.

9.2.3 AirProtocolLLRPCapabilities Parameter

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

AirProtocolLLRPCapabilities Parameter

Each air protocol's capabilities are expressed in a different LLRP Parameter. Each

protocol's air protocol capabilities parameter SHALL be referenced not more than once. The air protocol specific capabilities LLRP Parameters are defined in section 15.1.

9.2.4 RegulatoryCapabilities Parameter

This parameter carries the RF regulation specific attributes. They include regulatory standard, frequency band information, power levels supported, frequencies supported, and any air protocol specific values that are determined based on regulatory restriction.

The regulatory standard is encoded using two Integer fields, <Country Code, Communications standard> and it specifies the current operational regulatory mode of the device. This should not be used to reflect the ability to operate in regulatory environments which require configuration different from the current. This version of the LLRP protocol will have support for only the UHF band.

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

RegulatoryCapabilities Parameter

Country Code: Unsigned Short Integer. This field carries the numeric code of the country as specified in ISO 3166 [ISO3166]. 0 means unspecified.

Communications Standard: Unsigned Short Integer. This field carries the enumerations of the communications standard as specified below.

Possible Values:

Value	Communications Standard
-----	-----
0	Unspecified
1	US FCC Part 15
2	ETSI 302-208
3	ETSI 300-220
4	Australia LIPD 1W
5	Australia LIPD 4W
6	Japan: ARIB STD T89
7	Hong Kong: OFTA 1049
8	Taiwan: DGT LP0002
9	Korea: MIC Article 5-2
10-65535	Reserved for future use

UHFBandCapabilities: <UHFBandCapabilities Parameter> [optional]

Custom Extension Point List: List of <custom Parameter> [optional]

9.2.4.1 UHFBandCapabilities Parameter

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

UHFBandCapabilities Parameter

TransmitPowerTable: List of <TransmitPowerLevelTableEntry Parameter>

Frequency Information: <FrequencyInformation Parameter>

UHF_RFModeTable: List of LLRP Parameter.

Possible Values:

Each air protocol's UHF RF mode table is expressed as a different LLRP parameter. Each protocol SHALL be referenced not more than once. The air protocol's UHF RF mode table capabilities LLRP Parameters are defined in section 15.1.

1315 **9.2.4.1.1 TransmitPowerLevelTableEntry Parameter**

1316 This parameter specifies the index into the TransmitPowerLevelTable for a transmit
1317 power value. The transmit power is expressed in dBm*100 to allow fractional dBm
1318 representation and is the conducted power at the connector of the Reader.

1319 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1320 parameter.

TransmitPowerLevelTableEntry Parameter

Index: Integer

Possible Values: 0-255

Transmit power value: Signed short integer. Transmit power expressed in dBm*100 to allow fractional dBm representation.

1321 **9.2.4.1.2 FrequencyInformation Parameter**

1322 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1323 parameter.

Frequency Information Parameter

Hopping: Boolean

Freq Hop Info: Zero or more instances of <FrequencyHopTable Parameter>. This is transmitted only when Hopping = true.

Fixed Freq Info: At most one instance of <FixedFrequencyTable>. This is transmitted only when Hopping = false.

1324 **9.2.4.1.2.1 FrequencyHopTable Parameter**

1325 This parameter carries the frequency hop table parameters. This is used for Readers
1326 operating in regions with frequency hopping regulatory requirements. If the Reader is
1327 capable of storing multiple hop tables, the Reader may send all of them to the Client.
1328 Each hop table contains:

- 1329
- HopTableID which is the index of the frequency hop table returned by the Reader.

1330 • This is followed by a list of the frequencies (in Khz) in hop table order. The
1331 position of a frequency in the list is its ChannelIndex.

1332 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1333 parameter when operating in frequency hopping regulatory regions.

FrequencyHopTable Parameter

HopTableID : Integer

Possible Values: 0 - 255

Frequency Hop List: List of unsigned integers. Frequency in Khz.

1334 If multiple frequency hop tables are supported by the Reader, each table can be sent using
1335 a separate Frequency Hop Table Parameter.

1336 **9.2.4.1.2.2 FixedFrequencyTable Parameter**

1337 This parameter carries the fixed frequency list that can be used by the Reader. The
1338 position of a frequency in the list is its ChannelIndex.

1339 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1340 parameter when operating in fixed frequency regulatory regions.

Fixed Frequency Parameter

Frequency List: List of unsigned integers. Frequency in Khz.

1341 **10 Reader Operation (RO)**

1342 This section presents the messages and the parameters used by the Client for specifying
1343 RO.

1344 **10.1 Messages**

1345 **10.1.1 ADD_ROSPEC**

1346 An ADD_ROSPEC message communicates the information of a *ROSpec* to the Reader.
1347 LLRP supports configuration of multiple ROSpecs. Each ROSpec is uniquely identified
1348 using a ROSpecID, generated by the Client. The *ROSpec* starts at the Disabled state
1349 waiting for the ENABLE_ROSPEC message for the *ROSpec* from the Client, upon which
1350 it transitions to the Inactive state.

1351 The Client SHALL add a ROSpec in a Disabled State – i.e., CurrentState field in the
1352 ROSpec Parameter (section 10.2.1) SHALL be set to disabled. If the CurrentState value
1353 is different than disabled, an error SHALL be returned in the
1354 ADD_ROSPEC_RESPONSE (e.g. P_FieldError).

1355 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1356 message.

ADD_ROSPEC

ROSpec: ROSpec Parameter

10.1.2 ADD_ROSPEC_RESPONSE

This is the response by the Reader to an ADD_ROSPEC message. If all the parameters specified in the ADD_ROSPEC command are successfully set, then the success code is returned in the LLRPStatus parameter. If there is an error, the appropriate error code is returned in the LLRPStatus parameter.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

ADD_ROSPEC_RESPONSE

Response: LLRPStatus Parameter

10.1.3 DELETE_ROSPEC

This command is issued by the Client to the Reader. This command deletes the ROSpec at the Reader corresponding to ROSpecID passed in this message.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

DELETE_ROSPEC

ROSpecID: Unsigned Integer. The identifier of the ROSpec to delete. 0 indicates to delete all ROSpecs.

10.1.4 DELETE_ROSPEC_RESPONSE

This is the response by the Reader to a DELETE_ROSPEC command. If there was a ROSpec corresponding to the ROSpecID that the Reader was presently executing, and the Reader was successful in stopping that execution, then the success code is returned in the LLRPStatus parameter. If there is an error, the appropriate error code is returned in the LLRPStatus parameter.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

DELETE_ROSPEC_RESPONSE

Response: LLRPStatus Parameter

10.1.5 START_ROSPEC

This message is issued by the Client to the Reader. Upon receiving the message, the Reader starts the ROSpec corresponding to ROSpecID passed in this message, if the ROSpec is in the enabled state.

1381 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1382 message.

START_ROSPEC

ROSpecID: Unsigned Integer. The identifier of the ROSpec to start.

Possible Values: 0 is disallowed.

1383 **10.1.6 START_ROSPEC_RESPONSE**

1384 This is the response by the Reader to a START_ROSPEC command. If there was a
1385 ROSpec corresponding to the ROSpecID in the enabled state, and the Reader was able to
1386 start executing that ROSpec, then the success code is returned in the LLRPStatus
1387 parameter. If there is an error, the appropriate error code is returned in the LLRPStatus
1388 parameter.

1389 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1390 message.

START_ROSPEC_RESPONSE

Response: LLRPStatus Parameter

1391 **10.1.7 STOP_ROSPEC**

1392 This message is issued by the Client to the Reader. Upon receiving the message, the
1393 Reader stops the execution of the ROSpec corresponding to the ROSpecID passed in this
1394 message. STOP_ROSPEC overrides all other priorities and stops the execution. This
1395 basically moves the ROSpec's state to Inactive. This message does not delete the
1396 ROSpec.

1397 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1398 message.

STOP_ROSPEC

ROSpecID: Unsigned Integer. The identifier of the ROSpec to stop.

Possible Values: 0 is disallowed.

1399 **10.1.8 STOP_ROSPEC_RESPONSE**

1400 This is the response by the Reader to a STOP_ROSPEC command. If the Reader was
1401 currently executing the ROSpec corresponding to the ROSpecID, and the Reader was
1402 able to stop executing that ROSpec, then the success code is returned in the LLRPStatus
1403 parameter. If there is an error, the appropriate error code is returned in the LLRPStatus
1404 parameter.

1405 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1406 message.

STOP_ROSPEC_RESPONSE

Response: LLRPStatus Parameter

10.1.9 ENABLE_ROSPEC

This message is issued by the Client to the Reader. Upon receiving the message, the Reader moves the ROSpec corresponding to the ROSpecID passed in this message from the disabled to the inactive state.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

ENABLE_ROSPEC

ROSpecID: Unsigned Integer. The identifier of the ROSpec to enable. If set to 0, all ROSpecs are enabled.

10.1.10 ENABLE_ROSPEC_RESPONSE

This is the response by the Reader to a ENABLE_ROSPEC command. If there was a ROSpec corresponding to the ROSpecID, and the Reader was able to enable that ROSpec, then the success code is returned in the LLRPStatus parameter. If there is an error, the appropriate error code is returned in the LLRPStatus parameter.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

ENABLE_ROSPEC_RESPONSE

Response: LLRPStatus Parameter

10.1.11 DISABLE_ROSPEC

This message is issued by the Client to the Reader. Upon receiving the message, the Reader moves the ROSpec corresponding to the ROSpecID passed in this message to the disabled state.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

DISABLE_ROSPEC

ROSpecID: Unsigned Integer. The identifier of the ROSpec to disable. If set to 0, all ROSpecs are disabled.

10.1.12 DISABLE_ROSPEC_RESPONSE

This is the response by the Reader to a DISABLE_ROSPEC command. If there was a ROSpec corresponding to the ROSpecID, and the Reader was able to disable that

1429 ROSpec, then the success code is returned in the LLRPStatus parameter. If there is an
1430 error, the appropriate error code is returned in the LLRPStatus parameter.

1431 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1432 message.

DISABLE_ROSPEC_RESPONSE

Response: LLRPStatus Parameter

1433 **10.1.13 GET_ROSPECS**

1434 This is the request from the Client to the Reader to retrieve all the ROSpecs that have
1435 been configured at the Reader.

1436 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1437 message.

GET_ROSPECS

1438 **10.1.14 GET_ROSPECS_RESPONSE**

1439 This is the response by the Reader to a GET_ROSPECS command. If there are no
1440 ROSpecs configured at the Reader, the response is just the LLRPStatus parameter with
1441 the success code. Else, a list of ROSpec parameter is returned by the Reader, along with
1442 the success code in the LLRPStatus parameter.

1443 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1444 message.

GET_ROSPECS_RESPONSE

Status: LLRPStatus Parameter

Response: List of <ROSpec Parameter> that are in the order in which they are added.

1445 **10.2Parameters**

1446 **10.2.1 ROSpec Parameter**

1447 This parameter carries the information of the Reader inventory and survey operation.

1448 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1449 parameter.

ROSpec Parameter

ROSpecID: Unsigned Integer; 0 is an illegal ROSpecID for a ROSpec.

Priority: Integer. Lower numbered priority values are given higher priority.

Possible Values: 0-7.

ROBoundarySpec: ROBoundarySpec Parameter

ListOfSpecs: List of <AISpec Parameter> and/or <RFSurveySpec Parameter> and/or Custom Parameter.

ROReportSpec: ROReportSpec Parameter [optional] (Section 13.2.1)

CurrentState: Integer

Possible Values:

Value	Definition
-----	-----
0	Disabled
1	Inactive
2	Active

10.2.1.1 ROBoundarySpec Parameter

This parameter carries the lifetime of the command, ROStartTrigger and ROStopTrigger parameters.

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

ROBoundarySpec Parameter

ROSpecStartTrigger: ROSpecStartTrigger Parameter

ROSpecStopTrigger: ROSpecStopTrigger Parameter

10.2.1.1.1 ROSpecStartTrigger Parameter

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

ROSpecStartTrigger Parameter

ROSpecStartTriggerType: Integer

Possible Values:

Value	Definition
-----	-----
0	Null - No start trigger. The only way to start the ROSpec is with a START_ROSPEC from the Client.
1	Immediate
2	Periodic
3	GPI

PeriodicTriggerValue: PeriodicTriggerValue Parameter [Optional]. This parameter SHALL be present when ROSpecStartTriggerType = 2.

GPITriggerValue: GPITriggerValue Parameter [Optional]. This parameter SHALL be present when ROSpecStartTriggerType = 3.

10.2.1.1.1.1 PeriodicTriggerValue Parameter

Periodic trigger is specified using UTC time, offset and period.

For one-shot inventory, period is set to 0, and for periodic inventory operation period > 0.

1461 If UTC time is not specified, the first start time is determined as (time of message receipt
1462 + offset), else, the first start time is determined as (UTC time + offset). Subsequent start
1463 times = first start time + k * period (where, k > 0).
1464 If the Reader does not support UTC clock (as indicated by HasUTCClockCapability), and
1465 it receives the UTC time as part of the PeriodicTriggerValue parameter from the Client,
1466 the Reader SHALL return an error.
1467 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1468 parameter. Compliant Readers and Clients MAY implement the UTCTimestamp
1469 parameter.

PeriodicTriggerValue Parameter

UTC Time: <UTCTimestamp Parameter> [Optional]

Offset: Unsigned Integer. Time offset specified in milliseconds.

Period: Unsigned Integer. Time period specified in milliseconds

1470 **10.2.1.1.1.2 GPITriggerValue Parameter**

1471 This trigger is tied to an event on the General Purpose Input (GPI) of the Reader. The
1472 event is represented as a boolean type, and it is up to the internal implementation of the
1473 Reader to map exact physical event to a boolean type. For example, a 0 → 1 and a 1 → 0
1474 transition on an input pin of the Reader could be mapped to a boolean true and a boolean
1475 false event respectively.

1476 This trigger parameter has a timeout value field. The timeout is useful for specifying a
1477 fail-safe timeout when this trigger is used as a stop trigger. When the timeout is 0, it
1478 indicates that there is no timeout. When used as a start trigger, the timeout value SHALL
1479 be ignored.

1480 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1481 parameter. Readers that do not support GPIs SHALL return zero for numGPIs in the
1482 capabilities discovery. If the Client sets up the GPI trigger for such a Reader, the Reader
1483 SHALL send an error message for the ADD_ROSPEC message and not add the ROSpec.

GPITriggerValue Parameter

GPIPortNum: Unsigned Short Integer.

Possible Values: 1-65535. Zero is invalid.

GPIEvent: Boolean. The Boolean value that causes a GPI event to trigger.

Timeout: Unsigned Integer. Trigger timeout in milliseconds. If set to zero, it indicates there is no timeout.

1484 **10.2.1.1.2 ROSpecStopTrigger Parameter**

1485 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1486 parameter.

ROSpecStopTrigger Parameter

ROSpecStopTriggerType: Integer

Possible Values:

Value	Definition
-----	-----
0	Null – Stop when all AISpecs are done, or when Preempted, or with a STOP_ROSPEC from the Client.
1	Duration
2	GPI with a timeout value

DurationTriggerValue: Duration in milliseconds. This field is ignored when ROSpecStopTriggerType != 1.

GPITriggerValue: GPITriggerValue Parameter [Optional]. This parameter SHALL be present when ROSpecStopTriggerType = 2.

10.2.2 AISpec Parameter

This parameter defines antenna inventory operations.

Compliance Requirement: Compliant Readers and Clients SHALL implement this parameter.

AISpec Parameter

AISpecStopTrigger: <AISpecStopTrigger Parameter>

AntennaIDs: Short Array. If this set contains an antenna ID of zero, this AISpec will utilize all the antennas of the Reader.

InventoryParameterSpecs: <List of InventoryParameterSpec Parameter>

Custom Extension Point List: List of <custom Parameter> [Optional]

10.2.2.1 AISpecStopTrigger Parameter

This parameter defines the stop (i.e., terminating boundary) of an antenna inventory operation.

Compliance Requirement: Compliant Readers and Clients SHALL implement this parameter. If a Reader reports NumGPIS (see GPIO Capabilities Parameter) greater than zero, then the Reader SHALL support GPI Trigger.

AISpecStopTrigger Parameter

AISpecStopTriggerType: Integer

Possible Values:

Value	Definition
-----	-----
0	Null – Stop when ROSpec is done.
1	Duration
2	GPI with a timeout value
3	Tag observation

Duration Trigger: Unsigned Integer. Duration of AISpec in milliseconds. This field SHALL be ignored when AISpecStopTriggerType != 1.

GPI Trigger : GPITrigger value Parameter [Optional]. This field SHALL be present when AISpecStopTriggerType = 2.

TagObservation Trigger : TagObservation Trigger Parameter [Optional]. This field SHALL be present when AISpecStopTriggerType = 3.

1497 **10.2.2.1.1 TagObservationTrigger Parameter**

1498 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1499 parameter.

Tag ObservationTrigger Parameter

TriggerType: Integer

Possible Values:

Value	Modulation
-----	-----
0	Upon seeing N tag observations, or timeout
1	Upon seeing no more new tag observations for t ms, or timeout
2	N attempts to see all tags in the FOV, or timeout

NumberOfTags: Unsigned Short Integer. This field SHALL be ignored when TriggerType != 0.

NumberOfAttempts; Unsigned Short Integer. This field SHALL be ignored when TriggerType != 2.

T : Unsigned Short Integer. Idle time between tag responses in milliseconds. This field SHALL be ignored when TriggerType != 1.

Timeout : Unsigned Integer; Trigger timeout value in milliseconds. If set to zero, it indicates that there is no timeout.

1500 **10.2.2.2 InventoryParameterSpec Parameter**

1501 This parameter defines the inventory operation to be performed at all antennas specified
1502 in the corresponding AISpec. This parameter is composed of
1503 an InventoryParameterSpecID, a ProtocolID, and zero or more optional antenna
1504 configuration parameters. Antenna configurations for antennas not indicated by the
1505 AntennaIDs within the AISpec are ignored by the reader.

1506
1507 **Compliance Requirement:** Compliant Readers and Clients SHALL implement this
1508 parameter.

InventoryParameterSpec Parameter

InventoryParameterSpecID: Unsigned Short Integer. 0 is illegal.

ProtocolID: Integer. Enumeration based on Table 3.

AntennaConfiguration: List of <AntennaConfiguration Parameter> (Section 12.2.6) [Optional]

Custom Extension Point List: List of <Custom Parameter> [Optional]

10.2.3 RFSurveySpec Parameter

This parameter defines RF Survey operations. RF Survey is an operation during which the Reader performs a scan and measures the power levels across a set of frequencies at an antenna. This parameter defines the identifier of the antenna where this survey is to be performed, the duration of the survey operation (specified via stop trigger), and the range of frequencies to measure power levels of.

Compliance Requirement: Compliant Readers and Clients MAY implement this parameter.

RFSurveySpec Parameter

Antenna ID: Unsigned Short Integer.

Possible Values: 1 to N, where N is the maximum number of antennas supported by the Reader.

RFSurveySpecStopTrigger: RFSurveySpecStopTrigger parameter

StartFrequency: Unsigned Integer. The start (lower bound) frequency to survey specified in Khz.

EndFrequency: Unsigned Integer in Khz. The end (upper bound) frequency to survey specified in Khz.

Custom Extension Point List: List of <custom Parameter> [Optional]

10.2.3.1 RFSurveySpecStopTrigger Parameter

This parameter defines the stop trigger for RF Survey operations.

Compliance Requirement: Compliant Readers and Clients MAY implement this parameter.

RFSurveySpecStopTrigger Parameter

StopTriggerType: Integer

Possible Values:

Value	Definition
-----	-----
0	Null
1	Duration
2	N iterations through the frequency range

Duration: Unsigned Integer; The maximum duration of the RFSurvey operation specified in milliseconds. This field SHALL be ignored when StopTriggerType != 1. When StopTriggerType = 1, the value SHALL be greater than zero.

N: Unsigned Integer. The maximum number of iterations through the specified

frequency range. This field SHALL be ignored when StopTriggerType != 2. When StopTriggerType = 2, the value SHALL be greater than zero.

11 Access Operation

This section presents the messages and the parameters used by the Client for specifying access operation.

11.1 Messages

11.1.1 ADD_ACCESSSPEC

This command creates a new AccessSpec at the Reader. The *AccessSpec* starts at the Disabled state waiting for the ENABLE_ACCESSSPEC message for the *AccessSpec* from the Client, upon which it transitions to the Active state. The AccessSpecID is generated by the Client.

The Client SHALL add an AccessSpec in a Disabled State – i.e., CurrentState field in the AccessSpec Parameter (section 11.2.1) SHALL be set to false. If the CurrentState value is different than false, an error SHALL be returned in the ADD_ACCESSSPEC_RESPONSE (e.g. P_FieldError).

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

ADD_ACCESSSPEC

AccessSpec: AccessSpec parameter

11.1.2 ADD_ACCESSSPEC_RESPONSE

This is the response by the Reader to an ADD_ACCESSSPEC command. If the parameters passed in that ADD_ACCESSSPEC command were successfully accepted and set at the Reader, then the success code is returned in the LLRPStatus parameter. However, if the *AccessSpec* was not successfully created at the Reader, the Reader sends a LLRPStatus parameter describing the error in the message.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

ADD_ACCESSSPEC_RESPONSE

Response: LLRPStatus Parameter

11.1.3 DELETE_ACCESSSPEC

This command is issued by the Client to the Reader. The Reader deletes the AccessSpec corresponding to the AccessSpecId, and this AccessSpec will stop taking effect from the next inventory round.

1548 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1549 message.

DELETE_ACCESSSPEC

AccessSpecID : Unsigned Integer.

Possible Values: A value of zero is illegal.

1550 **11.1.4 DELETE_ACCESSSPEC_RESPONSE**

1551 This is the response by the Reader to a DELETE_ACCESSSPEC command. If there was
1552 an AccessSpec at the Reader corresponding to the AccessSpecID passed in the
1553 DELETE_ACCESSSPEC command, and the Reader was successful in deleting that
1554 AccessSpec, then the success code is returned in the LLRPStatus parameter. If there is an
1555 error, the appropriate error code is returned in the LLRPStatus parameter.

1556 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1557 message.

DELETE_ACCESSSPEC_RESPONSE

Response: LLRPStatus Parameter

1558 **11.1.5 ENABLE_ACCESSSPEC**

1559 This message is issued by the Client to the Reader. Upon receiving the message, the
1560 Reader moves the AccessSpec corresponding to the AccessSpecID in this message from
1561 the Disabled state to the Active state. The Reader executes this access-spec until it gets a
1562 DISABLE_ACCESSSPEC or a DELETE_ACCESSSPEC from the Client. The
1563 AccessSpec takes effect with the next (and subsequent) inventory rounds.

1564 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1565 message.

ENABLE_ACCESSSPEC

AccessSpecID: Unsigned Integer. If set to 0, all AccessSpecs are enabled.

1566 **11.1.6 ENABLE_ACCESSSPEC_RESPONSE**

1567 This is the response by the Reader to a START_ACCESSSPEC command. If there was
1568 an AccessSpec corresponding to the AccessSpecID, and the Reader was able to move that
1569 AccessSpec from the disabled to the active state, then the success code is returned in the
1570 LLRPStatus parameter. If there is an error, the appropriate error code is returned in the
1571 LLRPStatus parameter.

1572 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1573 message.

ENABLE_ACCESSSPEC_RESPONSE

Response: LLRPStatus Parameter

11.1.7 DISABLE_ACCESSSPEC

This message is issued by the Client to the Reader. Upon receiving the message, the Reader stops the execution of the AccessSpec corresponding to AccessSpecID in this message. This basically moves the AccessSpec's state to Disabled. This message does not delete the AccessSpec. The AccessSpec will stop taking effect from the next inventory round.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

DISABLE_ACCESSSPEC

AccessSpecID: Unsigned Integer. If set to 0, all AccessSpecs are disabled.

11.1.8 DISABLE_ACCESSSPEC_RESPONSE

This is the response by the Reader to a STOP_ACCESSSPEC command. If the Reader was currently executing the AccessSpec corresponding to the AccessSpecID, and the Reader was able to disable that AccessSpec, then the success code is returned in the LLRPStatus parameter. If there is an error, the appropriate error code is returned in the LLRPStatus parameter.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

DISABLE_ACCESSSPEC_RESPONSE

Response: LLRPStatus Parameter

11.1.9 GET_ACCESSSPECS

This is the request from the Client to the Reader to retrieve all the AccessSpecs that have been configured at the Reader.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

GET_ACCESSSPECS

11.1.10 GET_ACCESSSPECS_RESPONSE

This is the response by the Reader to a GET_ACCESSSPECS command. If there are no AccessSpecs configured at the Reader, the response is just the LLRPStatus parameter with the success code. Else, a list of <AccessSpecID, AccessSpec parameter> is returned by the Reader, along with the LLRPStatus parameter containing the success code. The order of the AccessSpecs listed in the message is normatively the order in which the AccessSpecs were created at the Reader.

1602 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1603 message.

GET_ACCESSSPECS_RESPONSE

Status: LLRPStatus Parameter

Response: List of <AccessSpec Parameter>. The ordering of the AccessSpecs in this list is the order in which the AccessSpecs were created at the Reader.

1604 **11.1.11 CLIENT_REQUEST_OP**

1605 This message is sent by the Reader to the Client upon executing a ClientRequestOpSpec
1606 OpSpec (section 11.2.1.2.1). This message carries the TagReportData (section 13.2.3)
1607 that contains information collected for the tag which includes singulation results and the
1608 results of OpSpecs executed till that point.

1609 **Compliance requirement:** Compliant Readers and Clients MAY implement this
1610 message.

CLIENT_REQUEST_OP

TagReport: <TagReportData Parameter> (Section 13.2.3)

1611 **11.1.12 CLIENT_REQUEST_OP_RESPONSE**

1612 This is the response by the Client to the Reader. This is in response to the
1613 CLIENT_REQUEST_OP sent by the Reader due to the execution of a
1614 ClientRequestOpSpec. This is a response to the CLIENT_REQUEST_OP message; thus,
1615 the messageID in this message is the messageID of the CLIENT_REQUEST_OP.

1616 **Compliance requirement:** Compliant Readers and Clients MAY implement this
1617 message. Readers that do not support ClientRequestOpSpec MAY ignore this message.

CLIENT_REQUEST_OP_RESPONSE

Response: ClientRequestResponse Parameter

1618

1619 **11.2Parameters**

1620 **11.2.1 AccessSpec Parameter**

1621 This parameter carries information of the Reader access operation.

1622 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1623 parameter.

AccessSpec Parameter

AccessSpecID: Unsigned Integer. 0 is illegal.

Antenna ID: Unsigned Short Integer. If 0, this spec is operational on all antennas.

ProtocolID: Integer.

Possible Values: Enumeration based on Table 3.

CurrentState: Boolean. This is the current state of the AccessSpec. false = Disabled, true = Active.

ROSpecID: Unsigned Integer. If 0, this spec is operational for all ROSpecs.

AccessSpecStopTrigger: AccessSpecStopTrigger Parameter

Access Command Operation: AccessCommand Parameter

AccessReportSpec: AccessReportSpec Parameter [Optional]

Custom Extension Point List: List of <custom Parameter> [Optional]

11.2.1.1 AccessSpecStopTrigger Parameter

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

AccessSpecStopTrigger Parameter

AccessSpecStopTriggerType: Integer

Possible Values:

Value	Definition
-----	-----
0	Null - No stop trigger defined.
1	Operation count

OperationCountValue: Unsigned Short Integer. A count to indicate the number of times this Spec is executed before it is deleted. If set to 0, this is equivalent to no stop trigger defined.

11.2.1.2 AccessCommand Parameter

This parameter defines the air protocol access-specific settings. It contains a TagSpec and an OpSpec Parameter. The TagSpec specifies the tag filters in terms of air protocol specific memory capabilities (e.g., memory banks, pointer and length). The OpSpec specifies all the details of the operations required for the air protocol specific access operation commands.

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

AccessCommand Parameter

TagSpec: LLRP Parameter

Possible Values:

Each air protocol's TagSpec parameter is expressed as a different LLRP Parameter. The air protocol specific TagSpec LLRP Parameters are defined in section 15.1. This field carries a single TagSpec parameter corresponding to the air protocol referenced by the ProtocolID in the AccessSpec Parameter.

OpSpec: List of LLRP Parameters

Possible Values:

Each parameter can be either an air protocol specific OpSpec LLRP Parameter or a <ClientRequestOpSpec Parameter>.

Regarding the air protocol specific OpSpec LLRP Parameter: Each air protocol's OpSpec parameter is expressed as a different LLRP Parameter. The air protocol specific OpSpec LLRP Parameters are defined in section 15.1. The list of OpSpecs in this field is comprised of OpSpec parameters corresponding to the air protocol referenced by the ProtocolID in the AccessSpec Parameter.

Custom Extension Point List: List of <Custom Parameter> [Optional]

1635

1636 In case there are multiple AccessSpecs that get matched during a TagSpec lookup, the
1637 Reader SHALL only execute the first enabled AccessSpec that matches, where the
1638 ordering of the AccessSpecs is the order in which the AccessSpecs were created by the
1639 Client.

1640 The order of execution of OpSpecs within an AccessSpec is the order in which the
1641 OpSpecs were set up in the AccessSpec. If an OpSpec execution fails, the Reader
1642 SHALL stop the execution of the AccessSpec.

1643 **11.2.1.2.1 ClientRequestOpSpec Parameter**

1644 This parameter is sent as part of the possible values for the AccessSpec OpSpec list. One
1645 or more ClientRequestOpSpec operations may be performed on a tag in succession.
1646 Upon executing a ClientRequestOpSpec Parameter, a Reader will immediately send the
1647 CLIENT_REQUEST_OP message to the Client. This CLIENT_REQUEST_OP message
1648 carries the TagReportData (section 13.2.3) that contains information collected for the tag
1649 which includes singulation results and the results of OpSpecs executed till that point.

1650 A global timeout is associated with this request. If the Client does not return a
1651 ClientRequestResponse within the *ClientRequestOpSpecTimeout* (LLRP Capabilities)
1652 period, or the AirProtocolOpSpec List is empty in the ClientRequestResponse, the
1653 execution of the AccessSpec is cancelled.

1654 **Compliance requirement:** Compliant Readers and Clients MAY implement this
1655 parameter. Readers that do not support ClientRequestOpSpec SHALL set
1656 SupportClientRequestOpSpec to false in LLRPCapabilities. If such a Reader receives an

1657 ADD_ACCESSSPEC with an AccessSpec that contained this OpSpec parameter, the
1658 Reader SHALL return an error for that message, and not add the AccessSpec.

ClientRequestOpSpec Parameter

OpSpecID: Unsigned Short Integer.

Possible Values: 0 is an illegal value.

1659 **11.2.2 ClientRequestResponse Parameter**

1660 This parameter describes the list of OpSpecs that the Reader has to execute on the tag for
1661 which a Client request was initiated. The AccessSpecID is the identifier of the
1662 AccessSpec that had the Client request; the EPC data is the singulated data of the tag for
1663 which this Client request was initiated. The AirProtocolOpSpec list contained in the
1664 ClientRequestResponse SHALL be processed as the next OpSpecs sent over the air
1665 interface. If the AirProtocolOpSpec List is empty, then the execution of the AccessSpec
1666 specified by AccessSpecID is cancelled.

1667 **Compliance requirement:** Compliant Readers MAY implement this parameter. Readers
1668 that do not support ClientRequestOpSpec MAY ignore this parameter.

ClientRequestResponse Parameter

AccessSpecID: Unsigned Integer. The ID of the AccessSpec that triggered this request.

EPCdata: <EPData Parameter>. The electronic product code of the RFID tag that triggered this request.

AirProtocolOpSpecList: List of LLRP OpSpec Parameter.

Possible Values:

Each air protocol's OpSpec parameter is expressed as a different LLRP Parameter. The air protocol specific OpSpec LLRP Parameters are defined in section 15.1. This field carries a list of OpSpec parameters corresponding to the air protocol referenced by ProtocolID in the AccessSpec that generated the Client request.

1669 **12 Reader Device Configuration**

1670 This section contains the messages and parameters for getting and setting configuration.

1671 **12.1 Messages**

1672 **12.1.1 GET_READER_CONFIG**

1673 This command is issued by the Client to get the current configuration information of the
1674 Reader. The Requested Data passed in the command represents the parameter(s) of
1675 interest to the Client that has to be returned by the Reader.

1676 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1677 message.

GET_READER_CONFIG

RequestedData : Integer

Possible Values:

Value	Requested Data
0	All
1	Identification
2	AntennaProperties
3	AntennaConfiguration
4	R0ReportSpec
5	ReaderEventNotificationSpec
6	AccessReportSpec
7	LLRPConfigurationStateValue
8	KeepaliveSpec
9	GPIPortCurrentState
10	GPOWriteData
11	EventsAndReports

AntennaID: Unsigned Short Integer. This field is ignored when RequestedData != 0 or 2 or 3. If the AntennaID is 0, get antenna information (AntennaProperties, AntennaConfiguration) for all antennas.

GPIPortNum: Unsigned Short Integer. This field is ignored when RequestedData != 0 or 9. If the GPIPortNum is 0, get GPI port current state for all GPI ports.

GPOPortNum: Unsigned Short Integer. This field is ignored when RequestedData != 0 or 10. If the GPOPortNum is 0, get GPO port current state for all GPO ports.

Custom Extension Point List: List of <custom Parameter> [Optional]

12.1.2 GET_READER_CONFIG_RESPONSE

This is the response by the Reader to the GET_READER_CONFIG message. The response is the LLRPStatus Parameter and the list of configuration parameters based on the RequestedData in GET_READER_CONFIG. If the GET_READER_CONFIG message did not have any errors, the success code is returned in the LLRPStatus parameter, and in addition the requested configuration parameters are returned. If there is an error, the appropriate error code is returned in the LLRPStatus parameter. The response contains at most one instance of each configuration parameter except for two cases, which are as follows:

- If RequestedData is 0, 2 or 3, and AntennaID is set to 0 in the GET_READER_CONFIG message, the Reader SHALL return one instance of AntennaProperties Parameter or AntennaConfiguration Parameter per requested antenna.
- If RequestedData is 0 or 9 (10), and GPIPortNum (GPOPortNum) is set to 0 in the GET_READER_CONFIG message, and, if the Reader supports GPI (GPO), the Reader SHALL return one instance of GPIPortCurrentState (GPOWriteData) Parameter per requested GPI Port (GPO Port).

1695 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1696 message.

GET_READER_CONFIG_RESPONSE

Status: LLRPStatus Parameter

Response Data: Set of LLRP Parameters.

Possible Values: The possible members are zero or more of

{< LLRPConfigurationStateValue Parameter>,
<ReaderEventNotificationSpec Parameter>,
<Antenna Properties Parameter>,
<Antenna Configuration Parameter>,
<ROReportSpec Parameter>,
<AccessReportSpec Parameter>,
<Identification Parameter>,
<KeepaliveSpec Parameter>,
<GPIPortCurrentState Parameter>,
<GPOWriteData Parameter>,
<EventsAndReports Parameter>
}

Custom Extension Point List: List of <custom Parameter> [Optional]

12.1.3 SET_READER_CONFIG

1697 This command is issued by the Client to the Reader. This command sets the Reader
1698 configuration using the parameters specified in this command. Values passed by the
1699 SET_READER_CONFIG SHALL apply for the duration of the LLRP connection, or
1700 until the values are changed by additional SET_READER_CONFIG messages.
1701

1702 For example, ROReportSpec defines the reporting of ROReport format and trigger for a
1703 ROSpec. ROReportSpec sent as part of SET_READER_CONFIG becomes the default
1704 ROReportSpec for the Reader. A ROReportSpec sent as part of ROSpec in the
1705 ADD_ROSPEC command overrides the default value for that ROSpec. However, in
1706 cases where there is no ROReportSpec specified in a ROSpec sent as part of
1707 ADD_ROSPEC, that particular ROSpec inherits the default ROReportSpec.

1708 The data field ResetToFactoryDefault informs the Reader to set all configurable values to
1709 factory defaults before applying the remaining parameters.

1710 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1711 message.

SET_READER_CONFIG

ResetToFactoryDefault: Boolean. If true, the Reader will set all configurable values to factory defaults before applying the remaining parameters.

Configuration Data: Set of LLRP Parameters

Possible Values: The possible members of the set are

{<ReaderEventNotificationSpec Parameter>,
<Antenna Properties Parameter>,
<Antenna Configuration Parameter>,
<ROReportSpec Parameter>,
<AccessReportSpec Parameter>,
<KeepaliveSpec Parameter>,
<GPOWriteData Parameter>,
<GPIPortCurrentState Parameter>,
<EventsAndReports Parameter>}

Custom Extension Point List: List of <custom Parameter> [Optional]

12.1.4 SET_READER_CONFIG_RESPONSE

This is the response by the Reader to a SET_READER_CONFIG command. If all the parameters specified in the SET_READER_CONFIG command are successfully set, then the success code is returned in the LLRPStatus parameter. If there is an error, the appropriate error code is returned in the LLRPStatus parameter.

Compliance requirement: Compliant Readers and Clients SHALL implement this message.

SET_READER_CONFIG_RESPONSE

Response: < LLRPStatus Parameter>

12.1.5 CLOSE_CONNECTION

This command is issued by the Client to the Reader. This command instructs the Reader to gracefully close its connection with the Client. Under normal operating conditions, a Client SHALL attempt to send this command before closing an LLRP connection. A Client should wait briefly for the Reader to respond with a CLOSE_CONNECTION_RESPONSE.

Upon receipt of this command, the Reader SHALL respond with the CLOSE_CONNECTION_RESPONSE message and it should then attempt to close the connection between the Reader and Client.

1728 Having executed a CLOSE_CONNECTION command, a Reader MAY persist its
1729 configuration state as defined by the ReaderConfigurationStateValue parameter specified
1730 in section 12.2.1.

1731 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1732 message.

CLOSE_CONNECTION

1733 12.1.6 CLOSE_CONNECTION_RESPONSE

1734 This is the response by the Reader to a CLOSE_CONNECTION command from the
1735 Client. Upon receiving a CLOSE_CONNECTION command, the Reader SHALL
1736 attempt to send this response to the Client. After attempting to send this response, the
1737 Reader SHALL close its connection with the Client.

1738 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1739 message.

CLOSE_CONNECTION_RESPONSE

Status: <LLRPStatus Parameter>

1740

1741 12.2 Parameters

1742 12.2.1 LLRPConfigurationStateValue Parameter

1743 This parameter, LLRPConfigurationStateValue, is a 32-bit value which represents a
1744 Reader's entire LLRP configuration state including: LLRP configuration parameters,
1745 vendor extension configuration parameters, ROSpecs, and AccessSpecs. A Reader
1746 SHALL change this value only:

- 1747 • Upon successful execution of any of the following messages:
 - 1748 o ADD_ROSPEC
 - 1749 o DELETE_ROSPEC
 - 1750 o ADD_ACCESSSPEC
 - 1751 o DELETE_ACCESSSPEC
 - 1752 o SET_READER_CONFIG
 - 1753 o Any CUSTOM_MESSAGE command that alters the reader's internal
1754 configuration.
- 1755 • Upon an automatically deleted AccessSpec due to completion of
1756 OperationCountValue number of operations (Section 11.2.1.1).

1757 A Reader SHALL not change this value when the CurrentState of a ROSpec or
1758 AccessSpec changes.

1759 The mechanism used to compute the LLRP configuration state value is implementation
1760 dependent. However, a good implementation will insure that there's a high probability
1761 that the value will change when the Reader's configuration state changes.

1762 It is expected that a Client will configure the Reader and then request the Reader's
1763 configuration state value. The Client will then save this state value. If this value does not
1764 change between two requests for it, then a Client may assume that the above components
1765 of the LLRP configuration have also not changed.

1766

1767 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1768 parameter. When requested by a Client, the Reader SHALL compute a state value based
1769 upon the Reader's current configuration state. Upon each request, the Reader SHALL
1770 return the same state value provided a Client has not altered the Reader's configuration
1771 state between requests. Aside from this requirement, the computation of the state value is
1772 implementation dependent.

LLRPConfigurationStateValue Parameter

LLRPConfigurationStateValue: Unsigned Integer

1773 12.2.2 Identification Parameter

1774 This parameter carries an identification parameter that is unique within the local
1775 administration domain. The identifier could be the Reader MAC address or EPC. The
1776 IDType defines the type of the identification value contained in this Parameter.

1777 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1778 parameter.

Identification Parameter

IDType: Integer

Possible Values:

IDType	ID
----	--
0	MAC address
1	EPC

Reader ID: Byte array. If IDType=0, the MAC address SHALL be encoded as EUI-64.[EUI64]

1779 12.2.3 GPOWriteData Parameter

1780 This parameter carries the data pertinent to perform the write to a general purpose output
1781 port.

1782 **Compliance requirement:** Compliant Readers and Clients MAY implement this
1783 parameter. Readers that do not support GPOs SHALL set NumGPOs in the
1784 GPIOCapabilities to zero. If such a Reader receives a SET_READER_CONFIG with

1785 GPOWriteData Parameter, the Reader SHALL return an error message and not process
1786 any of the parameters in that message.

GPOWriteData Parameter

GPO Port Number : Unsigned Short Integer. 0 is invalid.

GPO Data: Boolean. The state to output on the specified GPO port.

1787 **12.2.4 KeepaliveSpec Parameter**

1788 This parameter carries the specification for the keepalive message generation by the
1789 Reader. This includes the definition of the periodic trigger to send the keepalive message.

1790 **Compliance requirement**: Compliant Readers and Clients SHALL implement this
1791 parameter.

KeepaliveSpec Parameter

KeepaliveTriggerType: Integer

Possible Values:

Value	Definition
-----	-----
0	Null - No keepalives SHALL be sent by the Reader
1	Periodic

PeriodicTriggerValue: Integer. Time interval in milliseconds. This field is ignored when KeepaliveTriggerType is not 1.

1792 **12.2.5 AntennaProperties Parameter**

1793 This parameter carries a single antenna's properties. The properties include the gain and
1794 the connectivity status of the antenna. The antenna gain is the composite gain and includes
1795 the loss of the associated cable from the Reader to the antenna. The gain is represented in
1796 dBi*100 to allow fractional dBi representation.

1797 **Compliance requirement**: Compliant Readers and Clients MAY implement this
1798 parameter.

AntennaProperties Parameter

AntennaID: Unsigned Short Integer

Possible Values:

1 to N, where N is the maximum number of antennas supported by the device.

AntennaGain: Signed short integer. The gain of the antenna in dBi*100 (dB relative to Isotropic) to allow for fractional dBi representation.

AntennaConnected: Boolean. False = not connected, True = connected.

12.2.6 AntennaConfiguration Parameter

This parameter carries a single antenna's configuration and it specifies the default values for the parameter set that are passed in this parameter block. The scope of the default values is the antenna. The default values are used for parameters during an operation on this antenna if the parameter was unspecified in the spec that describes the operation.

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

AntennaConfiguration Parameter

Antenna ID: Unsigned Short Integer. If set to zero, this configuration applies to all the antennas.

RFReceiverSettings: <RFReceiver Parameter> [Optional]

RFTransmitterSettings: <RFTransmitter Parameter> [Optional]

AirProtocolInventoryCommandSettings: List of LLRP parameters. [Optional]

Possible Values:

Each air protocol's inventory command parameter is expressed as a different LLRP Parameter. The air protocol specific inventory command LLRP Parameters are defined in section 15.1. This field is a list of inventory command LLRP Parameters, one per air protocol, that the Client would like to use as the default inventory command setting for inventory operations using the air protocol on this antenna.

12.2.6.1 RFReceiver Parameter

This Parameter carries the RF receiver information. The Receiver Sensitivity defines the sensitivity setting at the receiver. The value is the index into the ReceiveSensitivityTable (section 9.2.1.1).

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

RFReceiver Parameter

ReceiverSensitivity: Unsigned Short Integer - an index into the ReceiveSensitivity Table (section 9.2.1.1)

12.2.6.2 RFTransmitter Parameter

This Parameter carries the RF transmitter information. The Transmit Power defines the transmit power for the antenna expressed as an index into the TransmitPowerTable (section 9.2.4.1.1). The HopTableID is the index of the frequency hop table to be used by the Reader (section 9.2.4.1.2.1) and is used when operating in frequency-hopping regulatory regions. This field is ignored in non-frequency-hopping regulatory regions. The ChannelIndex is the channel index in the FixedFrequencyTable to use during transmission (section 9.2.4.1.2.2) and is used when operating in non-frequency-hopping regulatory regions. This field is ignored in frequency-hopping regulatory regions.

1821 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1822 parameter.

RFTransmitter Parameter

Transmit Power: Unsigned Short Integer - an index into the Transmit Power table.

HopTableID : Unsigned Short Integer

ChannelIndex : Unsigned Short Integer. This is the index of the frequency to use.

1823 **12.2.6.3 GPIPortCurrentState Parameter**

1824 This Parameter carries the current configuration and state of a single GPI port. In a
1825 SET_READER_CONFIG message, this parameter is used to enable or disable the GPI
1826 port using the GPIConfig field; the GPIState field is ignored by the reader. In a
1827 GET_READER_CONFIG message, this parameter reports both the configuration and
1828 state of the GPI port.

1829
1830 When a ROSpec or AISpec is configured on a GPI-capable reader with GPI start and/or
1831 stop triggers, those GPIs must be enabled by the client with a SET_READER_CONFIG
1832 message for the triggers to function.

1833
1834 **Compliance requirement:** Compliant Readers and Clients MAY implement this
1835 parameter. Readers that do not support GPIs SHALL set NumGPIs in the
1836 GPIOCapabilities to zero. If such a Reader receives a GET_READER_CONFIG with a
1837 GPIPortCurrentState Parameter, the Reader SHALL return an error message and not
1838 process any of the parameters in that message.

GPIPortCurrentState Parameter

GPIPortNum: Unsigned Short Integer. Zero is illegal.

GPIConfig : Boolean (0 for disabled, 1 for enabled)

GPIState : Integer (ignored in SET_READER_CONFIG messages)

Possible Values:

Value	Definition
-----	-----
0	GPI state is low
1	GPI state is high
2	GPI state is unknown

1839 **12.2.6.4 EventsAndReports Parameter**

1840 This parameter controls the behavior of the Reader when a new LLRP connection is
1841 established. In a SET_READER_CONFIG message, this parameter is used to enable or
1842 disable the holding of events and reports upon connection using the
1843 HoldEventsAndReportsUponReconnect field. In a GET_READER_CONFIG message,
1844 this parameter reports the current configuration. If the
1845 HoldEventsAndReportsUponReconnect is true, the reader will not deliver any reports or
1846 events (except the ConnectionAttemptEvent) to the Client until the Client issues an

1847 ENABLE_EVENTS_AND_REPORTS message. Once the
1848 ENABLE_EVENTS_AND_REPORTS message is received the reader ceases its hold on
1849 events and reports for the duration of the connection.

1850

1851 **Compliance requirement:** Compliant Readers and Clients MAY implement this
1852 parameter.

EventsAndReports Parameter

HoldEventsAndReportsUponReconnect: Boolean. (False does not hold reports and events, True holds reports and events)

1853 13 Reports, Notifications and Keepalives

1854 This section describes the messages and parameters used in reports, event notifications
1855 and keepalives that are generated by the Reader and sent to the Client.

1856 The Reader SHALL send reports only when

- 1857 • A reporting trigger (ROReportTrigger or AccessReportTrigger) generates a report
1858 while a connection is open, or
- 1859 • In response to an explicit Client request (GET_REPORT or
1860 ENABLE_EVENTS_AND_REPORTS), or
- 1861 • A notification event occurs and the event is enabled.

1862 The triggers may be specified per ROSpec and AccessSpec using ROReportSpec and
1863 AccessReportSpec parameters. In a report, the Reader SHALL send new data (results of
1864 ROSpecs and/or AccessSpecs) acquired since the last report message. The tag report data
1865 generated by the AccessReport trigger SHALL NOT duplicate the tag report data
1866 generated by the ROReportTrigger, and vice-versa.

1867 13.1 Messages

1868 13.1.1 GET_REPORT

1869 This message is issued by the Client to the Reader to get the tag reports. In response to
1870 this message, the Reader SHALL return tag reports accumulated.

1871 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1872 message.

GET_REPORT

1873 13.1.2 RO_ACCESS_REPORT

1874 This message is issued by the Reader to the Client, and it contains the results of the RO
1875 and Access operations. The ROReportSpec and AccessReportSpec parameters define the
1876 contents and triggers for this message.

1877 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1878 message.

RO_ACCESS_REPORT

InventoryAccessReportData: List of <TagReportData Parameter> [Optional]

RFSurveyReportData: List of <RFSurveyReportData Parameter> [Optional]

Custom Extension Point List: List of <custom Parameter> [Optional]

1879 **13.1.3 KEEPALIVE**

1880 This message is issued by the Reader to the Client. This message can be used by the
1881 Client to monitor the LLRP-layer connectivity with the Reader. The Client configures the
1882 trigger at the Reader to send the Keepalive message. The configuration is done using the
1883 KeepaliveSpec parameter (section 12.2.4).

1884 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1885 message.

KEEPALIVE

1886 **13.1.4 KEEPALIVE_ACK**

1887 A Client SHALL generate a KEEPALIVE_ACK in response to each KEEPALIVE
1888 received by the reader.

1889 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1890 message.

KEEPALIVE_ACK

1891 **13.1.5 READER_EVENT_NOTIFICATION**

1892 This message is issued by the Reader to the Client whenever an event that the Client
1893 subscribed to occurs. The pertinent event data is conveyed using the
1894 ReaderEventNotificationData parameter.

1895 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1896 message.

READER_EVENT_NOTIFICATION

ReaderEventNotificationData: ReaderEventNotificationData Parameter

1897

1898 **13.1.6 ENABLE_EVENTS_AND_REPORTS**

1899 This message can be issued by the Client to the Reader after a LLRP connection is
1900 established. The Client uses this message to inform the Reader that it can remove its hold
1901 on event and report messages. Readers that are configured to hold events and reports on

1902 reconnection (See Section 12.2.6.4) respond to this message by returning the tag reports
1903 accumulated (same way they respond to GET_REPORT (See Section 13.1.1)).
1904 **Compliance requirement:** Compliant Readers and Clients MAY implement this
1905 message.

ENABLE_EVENTS_AND_REPORTS

1906

1907 13.2 Parameters

1908 13.2.1 ROResultSpec Parameter

1909 This Parameter carries the Reader inventory and RF survey reporting definition for the
1910 antenna. This parameter describes the contents of the report sent by the Reader and
1911 defines the events that cause the report to be sent.

1912 The ROResultTrigger field defines the events that cause the report to be sent.

1913 The TagReportContentSelector parameter defines the desired contents of the report. The
1914 ROResultTrigger defines the event that causes the report to be sent by the Reader to the
1915 Client.

1916 See section 13.2.6.1 for details about the order that reports are to be sent with respect to
1917 Reader event notifications.

1918 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1919 parameter.

ROResultSpec Parameter

ROResultTrigger: Integer

Possible Values:

Value	Definition
-----	-----
0	None
1	(Upon N TagReportData Parameters or End of AISpec) Or (End of RFSurveySpec) - N=0 is unlimited.
2	Upon N TagReportData Parameters or End of ROSpec - N=0 is unlimited.

N: Unsigned Short Integer. This is the number of TagReportData Parameters used in
ROResultTrigger = 1 and 2. If N = 0, there is no limit on the number of
TagReportData Parameters. This field SHALL be ignored when ROResultTrigger =
0.

ReportContents: <TagReportContentSelector Parameter>

Custom Extension Point List: List of <Custom Parameter> [Optional]

1920 **13.2.1.1 TagReportContentSelector Parameter**
1921 This parameter is used to configure the contents that are of interest in TagReportData. If
1922 enabled, the field is reported along with the tag data in the TagReportData.
1923 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1924 parameter.

TagReportContentSelector
EnableROSpecID: Boolean
EnableSpecIndex: Boolean
EnableInventoryParameterSpecID: Boolean
EnableAntennaID: Boolean
EnableChannelIndex: Boolean
EnablePeakRSSI: Boolean
EnableFirstSeenTimestamp: Boolean
EnableLastSeenTimestamp: Boolean
EnableTagSeenCount: Boolean
AirProtocolSpecificEPCMemorySelector: LLRP parameter.
Possible Values:
Each air protocol’s EPC memory selector parameter is expressed as a different LLRP Parameter. The air protocol specific EPC memory selector LLRP Parameters are defined in section 15.1. This field is the EPC memory selector LLRP Parameter corresponding to the air protocol referenced by the ProtocolID in the ROSpec that the ROResultSpec is part of.
EnableAccessSpecID: Boolean

1925 **13.2.2 AccessReportSpec Parameter**
1926 This parameter sets up the triggers for the Reader to send the access results to the Client.
1927 In addition, the Client can enable or disable reporting of ROSpec details in the access
1928 results.
1929 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1930 parameter.

AccessReportSpec
AccessReportTrigger: Integer
Possible Values:

Value	Definition
0	Whenever ROResult is generated for the RO that triggered the execution of this AccessSpec.

13.2.3 TagReportData Parameter

This report parameter is generated per tag per accumulation scope. The only mandatory portion of this parameter is the EPCData parameter. If there was an access operation performed on the tag, the results of the OpSpecs are mandatory in the report. The other sub-parameters in this report are optional. LLRP provides three ways to make the tag reporting efficient:

- (i) Allow parameters to be enabled or disabled via TagReportContentSelector (section 13.2.1.1) in TagReportSpec.
- (ii) If an optional parameter is enabled, and is absent in the report, the Client SHALL assume that the value is identical to the last parameter of the same type received. For example, this allows the Readers to not send a parameter in the report whose value has not changed since the last time it was sent by the Reader.
- (iii) Allow accumulation of tag reports. See next section for details of accumulation.

Compliance Requirement: Compliant Readers and Clients SHALL implement this parameter.

TagReportData Parameter

EPCData: <EPCData Parameter>

ROSpecID: <ROSpecID Parameter> [Optional]

SpecIndex: <SpecIndex Parameter> [Optional]

InventoryParameterSpecID: <InventoryParameterSpecID Parameter> [Optional]

AntennaID: <AntennaID Parameter> [Optional]

PeakRSSI: <PeakRSSI Parameter> [Optional]

ChannelIndex: <ChannelIndex Parameter> [Optional]

FirstSeenTimestampUTC: <UTCFirstSeenTimestamp Parameter> [Optional]

FirstSeenTimeStampUptime: <UptimeFirstSeenTimestamp Parameter> [Optional]

LastSeenTimestampUTC: <UTCLastSeenTimestamp Parameter> [Optional]

LastSeenTimeStampUptime: <UptimeLastSeenTimestamp Parameter> [Optional]

TagSeenCount: <TagSeenCount Parameter> [Optional]

AirProtocolTagData: LLRP Parameters (e.g., C1G2EPC-PC, C1G2EPC-CRC) [Optional]

Possible Values:

Each air protocol's AirProtocolTagData parameter is expressed as a different LLRP

Parameter. The air protocol specific AirProtocolTagData LLRP Parameters are defined in section 15.1. This field is the AirProtocolTagData LLRP Parameter corresponding to the air protocol referenced by the ProtocolID of the InventoryParameterSpec during whose execution this tag was observed.

AccessSpecID: <AccessSpecID Parameter> [Optional]

OpSpecResultList: List of LLRP parameters [Optional]

Possible Values of each LLRP Parameter: Air protocol specific OpSpecResult parameter or <ClientRequestOpSpecResult Parameter>.

Regarding the air protocol specific OpSpecResult parameter: Each air protocol's OpSpecResult parameter is expressed as a different LLRP Parameter. The air protocol specific OpSpecResult LLRP Parameters are defined in section 15.1. This field is a list of OpSpecResult LLRP Parameters corresponding to the air protocol referenced by the ProtocolID of the AccessSpec.

Custom Extension Point List: List of <Custom Parameter> [Optional]

13.2.3.1 Accumulation of TagReportData

A Reader MAY accumulate multiple tag reports into a single tag report.. If a Reader accumulates, the Reader SHALL follow the accumulation rules specified in this section. The following specifies the rules for accumulating multiple tag observations into a single TagReportData:

- EPCData:
 - o The Reader SHALL not accumulate tag reports that do not have the same EPCData value.
- OpSpecResultList:
 - o The Reader SHALL not accumulate tag reports that do not have the same value for the OpSpec results in the OpSpecResultList.
- SpecID, SpecIndex, InventoryParameterSpecID, AntennaID, AirProtocolTagData, AccessSpecID:
 - o These fields are optional, and their reporting can be enabled by the Client. If the Client has enabled one or more fields listed above, the Reader SHALL not accumulate tag reports that do not have the same value for all the enabled fields.
- FirstSeenTimestamp, LastSeenTimestamp, PeakRSSI, TagSeenCount, ChannelIndex
 - o These fields are optional, and their reporting can be enabled by the Client. If the field is enabled, the Reader sets the value of these fields as follows:
 - FirstSeenTimestamp: The Reader SHALL set it to the time of the first observation amongst the tag reports that get accumulated in the TagReportData.

- 1972 ▪ LastSeenTimestamp: The Reader SHALL set it to the time of the
- 1973 last observation amongst the tag reports that get accumulated in the
- 1974 TagReportData.
- 1975 ▪ PeakRSSI: The Reader SHALL set it to the maximum RSSI value
- 1976 observed amongst the tag reports that get accumulated in the
- 1977 TagReportData.
- 1978 ▪ ChannelIndex: The Reader MAY set it to the index of the first
- 1979 channel the tag was seen.
- 1980 ▪ TagSeenCount: The Reader SHALL set it to the number of tag
- 1981 reports that get accumulated in the TagReportData.

1982 **13.2.3.2 EPCData Parameter**

1983 This parameter carries the EPC identifier information.

1984 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1985 parameter.

EPCData Parameter

EPC: Bit array

1986 **13.2.3.3 ROSpecID Parameter**

1987 This parameter carries the ROSpecID information.

1988 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1989 parameter.

ROSpecID Parameter

ROSpecID: Unsigned Integer

1990 **13.2.3.4 SpecIndex Parameter**

1991 This parameter carries the SpecIndex information. The SpecIndex indicates the item
1992 within the ROSpec that was being executed at the time the tag was observed.

1993 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1994 parameter.

SpecIndex Parameter

SpecIndex: Unsigned Short Integer

1995 **13.2.3.5 InventoryParameterSpecID Parameter**

1996 This parameter carries the InventoryParameterSpecID information.

1997 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
1998 parameter.

InventoryParameterSpecID Parameter

InventoryParameterSpecID: Unsigned Short Integer

1999 **13.2.3.6 AntennaID Parameter**

2000 This parameter carries the AntennaID information.

2001 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2002 parameter.

AntennaID Parameter

AntennaID: Unsigned Short Integer

2003 **13.2.3.7 PeakRSSI Parameter**

2004 This parameter carries the PeakRSSI information.

2005 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2006 parameter.

PeakRSSI Parameter

PeakRSSI: Signed Integer. The peak received power of the EPC backscatter in dBm.

Possible Values:

-128 to +127.

2007 **13.2.3.8 ChannelIndex Parameter**

2008 This parameter carries the ChannelIndex information.

2009 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2010 parameter.

ChannelIndex Parameter

ChannelIndex: Unsigned Integer

Possible Values: 0 to 255.

2011 **13.2.3.9 FirstSeenTimestampUTC Parameter**

2012 This parameter carries the FirstSeenTimestamp information in UTC.

2013 **Compliance requirement:** Compliant Readers and Clients that have UTC clocks
2014 SHALL implement this parameter.

FirstSeenTimestampUTC Parameter

Microseconds: Unsigned long Integer. This is the time elapsed since the Epoch (00:00:00 UTC, January 1, 1970) measured in microseconds.

2015 **13.2.3.10 FirstSeenTimestampUptime Parameter**
2016 This parameter carries the FirstSeenTimestamp information in Uptime.
2017 **Compliance requirement:** Compliant Readers and Clients that do not have UTC clocks
2018 SHALL implement this parameter. Compliant Readers and Clients that have UTC clocks
2019 MAY implement this parameter.

FirstSeenTimestampUptime Parameter

Microseconds: Unsigned long Integer. This is the time elapsed since boot, measured in microseconds.

2020 **13.2.3.11 LastSeenTimestampUTC Parameter**
2021 This parameter carries the LastSeenTimestamp information in UTC.
2022 **Compliance requirement:** Compliant Readers and Clients that have UTC clocks
2023 SHALL implement this parameter.

LastSeenTimestampUTC Parameter

Microseconds: Unsigned long Integer. This is the time elapsed since the Epoch (00:00:00 UTC, January 1, 1970) measured in microseconds.

2024 **13.2.3.12 LastSeenTimestampUptime Parameter**
2025 This parameter carries the LastSeenTimestamp information in Uptime.
2026 **Compliance requirement:** Compliant Readers and Clients that do not have UTC clocks
2027 SHALL implement this parameter. Compliant Readers and Clients that have UTC clocks
2028 MAY implement this parameter.

LastSeenTimestampUptime Parameter

Microseconds: Unsigned long Integer. This is the time elapsed since boot, measured in microseconds.

2029 **13.2.3.13 TagSeenCount Parameter**
2030 This parameter carries the tag seen count information. If TagSeenCount > 65535 for the
2031 report period, the reader SHALL report 65535.
2032 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2033 parameter.

TagSeenCount Parameter

Count: Unsigned Short Integer

2034 **13.2.3.14 ClientRequestOpSpecResult Parameter**
2035 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2036 parameter.

ClientRequestOpSpecResult Parameter

OpSpecID: Unsigned Short Integer. 0 is illegal.

13.2.3.15 AccessSpecID Parameter

This parameter carries the AccessSpecID information.

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

AccessSpecID Parameter

AccessSpecID: Unsigned Integer

13.2.4 RFSurveyReportData Parameter

This describes the content of the RF Survey Report.

Compliance requirement: Compliant Readers and Clients MAY implement this parameter.

RFSurveyReportData Parameter

ROSpecID: <ROSpecID Parameter> [Optional]

SpecIndex: <SpecIndex Parameter> [Optional]

FrequencyPowerLevelList: List of <FrequencyRSSILevelEntry Parameter>

Custom Extension Point List: List of <custom Parameter> [Optional]

13.2.4.1 FrequencyRSSILevelEntry Parameter

Compliance requirement: Compliant Readers and Clients MAY implement this parameter.

FrequencyRSSILevelEntry Parameter

Timestamp: Either <UTCTimestamp Parameter> or <Uptime Parameter>

Frequency: Unsigned Integer. The frequency on which the measurement was taken, specified in kHz.

Bandwidth: Unsigned Integer. The measurement bandwidth of the measurement in kHz.

Average RSSI: Integer in dBm. The average power level observed at this frequency.

Possible Values:

-128 to + 127

Peak RSSI: Integer in dBm. The peak power level observed at this frequency.

Possible Values:

-128 to + 127

2048 **13.2.5 ReaderEventNotificationSpec Parameter**
2049 This parameter is used by the Client to enable or disable notification of one or more
2050 Reader events. Notification of buffer overflow events and connection events
2051 (attempt/close) are mandatory, and not configurable.
2052 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2053 parameter.

ReaderEventNotificationSpec Parameter
EventNotificationSpecTable: List of <EventNotificationState Parameter>

2054 **13.2.5.1 EventNotificationState Parameter**
2055 This parameter is used to enable or disable notification of a single Reader event type.
2056 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2057 parameter.

EventNotificationState Parameter
EventType:
Possible Values:

Value	Definition
-----	-----
0	Upon hopping to next channel (e.g., in FCC regulatory region)
1	GPI event
2	ROSpec event (start/end/preempt)
3	Report buffer fill warning
4	Reader exception event
5	RFSurvey event (start/end)
6	AISpec event (end)
7	AISpec event (end) with singulation details
8	Antenna event (disconnect/connect)

NotificationState: Boolean; enable = true, disable = false.

2058 **13.2.6 ReaderEventNotificationData Parameter**
2059 This parameter describes the contents of the event notification sent by the Reader, and
2060 defines the events that cause the notification to be sent. Event notification messages may
2061 be sent by the Reader due to connection establishment/closing event, critical events such
2062 as hopping, fault-detection in a Reader functional block, buffer overflow, due to the
2063 activation of a Reader accessory trigger input (e.g. motion detection), or due to
2064 performance monitoring events such as abnormalities in the RF environment.
2065 Timestamp is the time that the events reported occurred.
2066 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2067 parameter.

ReaderEventNotificationData Parameter

Timestamp: Either <UTCTimestamp Parameter> or <Uptime Parameter>

Events: List of events.

Possible Values: The possible members of the list are

```
{  
    <HoppingEventParameter>,  
    <GPiEvent Parameter>,  
    <ROSpecEvent Parameter>,  
    <ReportBufferLevelWarningEvent Parameter>,  
    <ReportBufferOverflowErrorEvent Parameter>,  
    <ReaderExceptionEvent Parameter>,  
    <RFSurveyEvent Parameter>,  
    <AISpecEvent Parameter>,  
    <AntennaEvent Parameter>,  
    <ConnectionAttemptEvent Parameter>,  
    <ConnectionCloseEvent Parameter>  
}
```

Custom Extension Point List: List of <custom Parameter> [optional]

13.2.6.1 Requirements for Ordering of Event Reporting

LLRP assumes a reliable stream transport mechanism. Messages sent through LLRP will arrive in the order that they were sent over the transport and binding utilized. Status events within the same message SHALL be ordered chronologically.

Status events delivered by reader event notifications are useful, especially in conjunction with the tag report data. The following describes the requirements of the reader event notifications ordering with respect to the ordering of tag reports and Reader Event Notifications.

The following requirements are made on the ordering of Event Parameters with respect to each other and to tag report Parameters. These statements apply if the respective status events and report triggers are enabled.

If the start of an ROSpec is triggered by a GPI, the GPiEvent Parameter SHALL be sent before the ROSpecEvent Parameter signaling the start of the ROSpec.

If the end of an ROSpec is triggered by a GPI, the GPiEvent Parameter SHALL be sent before the ROSpecEvent Parameter signaling the end of the ROSpec.

2083 If an ROSpec contains one or more AISpecs, the ROSpecEvent parameter signaling the
2084 end of an ROSpec SHALL be sent after the AISpecEvent Parameter signaling the end of
2085 the last AISpec within that ROSpec.

2086 If one ROSpec pre-empts another ROSpec, the ROSpecEvent parameter signaling the
2087 preemption of the first ROSpec SHALL be sent before the ROSpecEvent parameter
2088 signaling the start of the next ROSpec.

2089 Tag data received during an ROSpec execution SHALL be sent between the
2090 ROSpecEvent parameter signaling the start of the ROSpec and the ROSpecEvent
2091 parameter signaling the end or preemption of the ROSpec if the ROReportTrigger is not
2092 set to 'None'.

2093 Tag data received during an AISpec execution SHALL be sent before the AISpecEvent
2094 Parameter signaling the end of the AISpec if the ROReportTrigger is not 'None' or 'end
2095 of RO Spec'

2096 Tag data received during the time on a channel SHALL be sent after the HoppingEvent
2097 parameter that announced this channel and before the next HoppingEvent parameter
2098 when the ROReportTrigger is not 'None' and N=1.

2099 **13.2.6.2 HoppingEvent Parameter**

2100 A Reader reports this event every time it hops frequency.

2101 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2102 parameter.

HoppingEvent Parameter

HopTableID: Unsigned Short Integer

NextChannelIndex: Unsigned Short Integer. This is the index of the next channel to which the Reader is going to hop to. The channel Ids are listed in the Frequency Hop Table.

2103 **13.2.6.3 GPIEvent Parameter**

2104 A reader reports this event every time an enabled GPI changes GPIState.

2105 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2106 parameter.

GPIEvent Parameter

GPIPortNumber: Unsigned Short Integer

GPIEvent: Boolean – True/False.

2107 **13.2.6.4 ROSpecEvent Parameter**

2108 This parameter carries the ROSpec event details. The EventType could be start or end of
2109 the ROSpec.

2110 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2111 parameter.

ROSpecEvent Parameter

ROSpecID: Unsigned Integer. This is the ID of the ROSpec that started, ended or got preempted.

EventType: Integer

Possible Values:

Value	Definition
-----	-----
0	Start of ROSpec
1	End of ROSpec
2	Preemption of ROSpec

PreemptingROSpecID: Integer. This field is ignored when EventType != 2. This field carries the ID of the preempting ROSpec.

2112 **13.2.6.5 ReportBufferLevelWarningEvent Parameter**

2113 A Reader can warn the Client that the Reader's report buffer is filling up. A Client can
2114 act upon this warning by requesting report data from the Reader, thereby freeing the
2115 Reader's report memory resources.

ReportBufferLevelWarningEvent Parameter

ReportBufferPercentageFull: Integer

Possible Values: 0-100

2116
2117 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2118 parameter. A Reader MAY send a report buffer level warning event whenever the
2119 Reader senses that its report memory resources are running short. The buffer level at
2120 which a warning is reported is Reader implementation dependent. A Client MAY act
2121 upon a report buffer level warning event by requesting report data from the Reader and
2122 thereby free report memory resources in the Reader.

2123 **13.2.6.6 ReportBufferOverflowErrorEvent Parameter**

2124 A Reader reports a buffer overflow event whenever report data is lost due to lack of
2125 memory resources.

ReportBufferOverflowErrorEvent Parameter

2126 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2127 parameter. A Reader SHALL report a buffer overflow event whenever report data is lost
2128 due to lack of memory resources.

2129 **13.2.6.7 ReaderExceptionEvent Parameter**

2130 The reader exception status event notifies the client that an unexpected event has
2131 occurred on the reader. Optional parameters provide more detail to the client as to the
2132 nature and scope of the event.

2133 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2134 parameter.

ReaderExceptionEvent Parameter

ROSpecID: <ROSpecID Parameter> [Optional]

SpecIndex: <Spec Index Parameter> [Optional]

InventoryParameterSpecID: <InventoryParameterSpecID Parameter> [Optional]

AntennaID: <AntennaID Parameter> [Optional]

AccessSpecID: <AccessSpecID Parameter> [Optional]

OpSpecID: <OpSpecID Parameter> [Optional]

Message: UTF-8 String

Custom Extension Point List: List of <custom Parameter> [Optional]

2135 **13.2.6.7.1 OpSpecID Parameter**

2136 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2137 parameter.

OpSpecID Parameter

OpSpecId: Unsigned Short Integer. 0 is illegal.

2138 **13.2.6.8 RFSurveyEvent Parameter**

2139 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2140 parameter.

RFSurveyEvent Parameter

ROSpecID: Unsigned Integer. The identifier of the ROSpec that contains the RFSurveySpec.

SpecIndex: Unsigned Short Integer. The index of the spec in the ROSpec.

EventType: Integer

Possible Values:

Value	Definition
-----	-----
0	Start of RFSurvey
1	End of RFSurvey

2141 **13.2.6.9 AISpecEvent Parameter**
2142 This parameter carries the AISpec event details. The EventType is the end of the AISpec.
2143 When reporting the end event, the AirProtocolSingulationDetails MAY be reported if it is
2144 supported by the Reader and EventType of 7 has been enabled (Section 13.2.5.1).
2145 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2146 parameter.

AISpecEvent Parameter

ROSpecID: Unsigned Integer. The identifier of the ROSpec that contains the AISpec.

SpecIndex: Unsigned Short Integer. The index of the spec in the ROSpec.

EventType: Integer

Possible Values:

Value	Definition
-----	-----
0	End of AISpec

AirProtocolSingulationDetails: LLRP parameter [Optional]

Possible Values:

Each air protocol's AirProtocolSingulationDetails parameter is expressed as a different LLRP Parameter. The air protocol specific AirProtocolSingulationDetails LLRP Parameters are defined in section 15.1. This field is the AirProtocolSingulationDetails LLRP Parameter corresponding to the air protocol referenced by the ProtocolID of the InventoryParameterSpec upon whose execution completion this event report was generated.

2147 **13.2.6.10 AntennaEvent Parameter**
2148 This event is generated when the Reader detects that an antenna is connected or
2149 disconnected.
2150 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2151 parameter.

AntennaEvent Parameter

AntennaID: Unsigned Short Integer

EventType: Integer

Possible Values:

Value	Definition
-----	-----
0	Antenna disconnected
1	Antenna connected

2152 **13.2.6.11 ConnectionAttemptEvent Parameter**

2153 This status report parameter establishes Reader connection status when the Client or
2154 Reader initiates a connection. See section 18.1, TCP Transport, for more details
2155 regarding the use of this report.

2156 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2157 parameter.

ConnectionAttemptEvent Parameter	
Status: Integer	
Possible Values:	
Value	Definition
-----	-----
0	Success
1	Failed (a Reader initiated connection already exists)
2	Failed (a Client initiated connection already exists)
3	Failed (any reason other than a connection already exists)
4	Another connection attempted

2158 **13.2.6.12 ConnectionCloseEvent Parameter**

2159 This status report parameter informs the Client that, unsolicited by the Client, the Reader
2160 will close the connection between the Reader and Client. Before the Reader closes a
2161 connection with the Client that is not solicited by the Client, the Reader SHALL first
2162 attempt to send a READER_EVENT_NOTIFICATION containing this parameter to the
2163 Client.

2164 Once the Reader sends this event to the Client, the Reader SHALL close the connection
2165 to the Client. This is also to say that, once the Reader sends this event, the Reader
2166 SHALL send no additional messages to the Client and the Reader SHALL ignore any
2167 messages received from the Client until another new connection is established.

2168 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2169 parameter.

ConnectionCloseEvent Parameter

2170
2171 **14 Errors**

2172 This section describes the errors that are solely based on LLRP protocol message parsing.
2173 The Reader SHALL discard the message if there is at least one error in the message, or
2174 cannot be fully processed. In addition, no portion of the message containing an error
2175 SHALL be executed by the Reader. In case the message has one or more errors, the
2176 Reader SHALL return at least one error parameter for one of the errors. The Reader
2177 MAY return more than one error parameter, one for each error. The errors are conveyed
2178 using a combination of ‘generic error codes’, a pointer to the culprit parameter/field, and
2179 a description of the error encoded as a string of UTF-8 characters.

2180 Typically the errors in the LLRP defined messages are conveyed inside of the responses
2181 from the Reader. However, in cases where the message received by the Reader contains
2182 an unsupported message type, or a CUSTOM_MESSAGE with unsupported parameters
2183 or fields, the Reader SHALL respond with the ERROR_MESSAGE.

2184 When a Reader or Client receives a command or notification with a version that is not
2185 supported, the receiver SHALL send an ERROR_MESSAGE in reply consisting of: A
2186 version that is the same as the received message, the message ID that matches the
2187 received message, and an LLRPStatusParameter with the ErrorCode set to
2188 M_UnsupportedVersion. This message SHALL contain no sub-parameters (such as Field
2189 Error, Parameter Error).

2190 Readers and Clients SHALL not respond to an ERROR_MESSAGE.

2191 **14.1 Messages**

2192 **14.1.1 ERROR_MESSAGE**

2193 This message is issued by the Reader to the Client, and it contains the LLRPStatus
2194 parameter that describes the error in the message.

2195 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2196 message.

ERROR_MESSAGE

Error: <LLRPStatus Parameter>

2197 **14.2 Parameters**

2198 First, the error codes are presented, and then later, error parameters are presented that
2199 identify the culprit field in the message.

2200 **14.2.1 LLRP Status Codes**

2201 Status can be a success or one of the error conditions. This section lists a set of generic
2202 error conditions that, in combination with the identifier of the culprit field, conveys the
2203 error condition. The codes are broken into four scopes: message, parameter, field and
2204 device. The device code indicates that the error is in the Reader device rather than the
2205 message, parameter or field.

2206

StatusCode	Name	Scope	Description
0	M_Success	Message	This code SHALL indicate that the message was received and processed successfully.
100	M_ParameterError		This code SHALL indicate that an error occurred with a parameter of this message.

101	M_FieldError		This code SHALL indicate that an error occurred with a field of this message.
102	M_UnexpectedParameter		This code SHALL indicate that an unexpected parameter was received with this message.
103	M_MissingParameter		This code SHALL indicate that a required parameter was missing from this message.
104	M_DuplicateParameter		This code SHALL indicate that a parameter, for which there must only be one instance at the Reader, was seen more than once in this message.
105	M_OverflowParameter		This code SHALL indicate that the maximum number of instances of the parameter has been exceeded at the Reader.
106	M_OverflowField		This code SHALL indicate that the maximum number of instances of the field has been exceeded at the Reader.
107	M_UnknownParameter		This code SHALL indicate that an unknown parameter was received in the message.
108	M_UnknownField		This code SHALL indicate that the field is unknown or not found at the Reader.
109	M_UnsupportedMessage		This code SHALL indicate that an unsupported message type was received.
110	M_UnsupportedVersion		This code SHALL indicate that the LLRP version in the received message is not supported by the Reader.
111	M_UnsupportedParameter		This code MAY indicate that the Parameter in the received message is not supported by the Reader.
200	P_ParameterError	Parameter	This code SHALL indicate that an error occurred with a parameter of this parameter.
201	P_FieldError		This code SHALL indicate that an error occurred with a field of this parameter.
202	P_UnexpectedParameter		This code SHALL indicate that an unexpected parameter was

			received with this message.
203	P_MissingParameter		This code SHALL indicate that a required parameter was missing from this parameter.
204	P_DuplicateParameter		This code SHALL indicate that a parameter, for which there must only be one instance, was seen more than once in this parameter.
205	P_OverflowParameter		This code SHALL indicate that the maximum number of instances of the parameter has been exceeded at the Reader.
206	P_OverflowField		This code SHALL indicate that the maximum number of instances of the field has been exceeded at the Reader.
207	P_UnknownParameter		This code SHALL indicate that an unknown parameter was received with this message.
208	P_UnknownField		This code SHALL indicate that the field is unknown or not found at the Reader.
209	P_UnsupportedParameter		This code SHALL indicate that an unsupported parameter was received.
300	A_Invalid	Field	This code SHALL indicate that the field value was considered invalid for a non specific reason. An example is a message with invalid SpecID for a ROSpec or AccessSpec.
301	A_OutOfRange		This code SHALL indicate that the field value did not fall within an acceptable range.
401	R_DeviceError	Reader	This code MAY indicate that there is a problem on the Reader rather than with a message, parameter, or field.

14.2.2 LLRPStatus Parameter

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

LLRPStatus Parameter

StatusCode: Integer.

Possible Values:

See the error code table (section 14.2.1) for possible values within the Message, Parameter or Field scope.

FieldError: <FieldError Parameter> [Optional]

ParameterError: <ParameterError Parameter> [Optional]

ErrorDescription: UTF-8 String

2210 **14.2.2.1 FieldError Parameter**

2211 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2212 parameter.

FieldError Parameter

FieldNum: Integer. Field number for which the error applies. The fields are numbered after the order in which they appear in the parameter or message body.

Possible Values:

0-65535

ErrorCode: Integer.

Possible Values:

See the error code table (section 14.2.1) for possible values within the Argument scope.

2213 **14.2.2.2 ParameterError Parameter**

2214 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2215 parameter.

ParameterError Parameter

ParameterType: Integer. The parameter type that caused this error.

Possible Values:

0 - 1023

ErrorCode: Integer.

Possible Values:

See the error code table (section 14.2.1) for possible values within the Parameter scope.

FieldError: <FieldError Parameter> [Optional]

ParameterError: <ParameterError Parameter> [Optional]

2216

15 Air Protocol Specific Parameters

For each air protocol supported by LLRP, the following subsection, 15.1, provides a table cross-referencing LLRP parameters and their corresponding air protocol specific parameters. All LLRP air protocol specific parameters are specified in the next subsection, 15.2.

15.1 LLRP Air Protocol Cross-Reference Tables

Within this section there is a separate subsection for each air protocol specified by LLRP. Each air protocol subsection includes a table cross-referencing LLRP parameters and their corresponding air protocol specific parameters.

Support for a new air protocol can be added to LLRP by adding new subsections to this section e.g., 15.1.2 and 15.1.3.

15.1.1 Class-1 Generation-2 (C1G2) Air Protocol

The Class-1 Generation-2 (C1G2) Air Protocol is specified by the EPCglobal Class-1 Generation-2 UHF RFID Protocol v1.1.0 specification.

The following table cross-references LLRP parameters to C1G2 air protocol specific parameters.

LLRP Parameter (Section #)	C1G2 Parameter (Section #)
AirProtocolLLRPCapabilities (9.2.3)	C1G2LLRPCapabilities 15.2.1.1.1
UHF_RFModeTable (9.2.4.1)	UHFC1G2RFModeTable (15.2.1.1.2)
AirProtocolInventoryCommandSettings (12.2.6)	C1G2InventoryCommand (15.2.1.2.1)
TagSpec (11.2.1.2)	C1G2TagSpec (15.2.1.3.1)
OpSpec (11.2.1.2)	C1G2OpSpec (15.2.1.3.2)
AirProtocolOpSpecList (11.2.2)	C1G2OpSpec (15.2.1.3.2)
AirProtocolSpecificEPCMemorySelector (13.2.1.1)	C1G2EPCMemorySelector (15.2.1.5.1)
AirProtocolTagData (13.2.3)	C1G2PC and C1G2CRC (15.2.1.5.2, 15.2.1.5.3)
AirProtocolSingulationDetails (13.2.6.9)	C1G2SingulationDetails (15.2.1.5.4)
Op Spec Results (13.2.3)	C1G2OpSpecResult (15.2.1.5.5)

2233 **15.2 LLRP Air Protocol Specific Parameters**
2234 Within this section there is a separate subsection for each air protocol specified by LLRP.
2235 Each air protocol subsection includes a definition for each air protocol specific
2236 parameter. Section 15.1 above cross-references LLRP parameters to the air protocol
2237 specific parameters specified in this section.
2238 Support for a new air protocol can be added to LLRP by adding new subsections to this
2239 section e.g., 15.2.2 and 15.2.3.

2240 **15.2.1 Class-1 Generation-2 (C1G2) Air Protocol**

2241 The Class-1 Generation-2 (C1G2) Air Protocol is specified by EPCglobal Class-1
2242 Generation-2 UHF RFID Protocol v1.1.0 specification.

2243 The following subsections specify LLRP air protocol specific parameters. These
2244 subsections are partitioned to correlate with major sections of the LLRP specification:

- 2245 - Reader Device Capabilities
- 2246 - Inventory Operation
- 2247 - Access Operation
- 2248 - Reader Device Configuration
- 2249 - Reports

2250 **15.2.1.1 Reader Device Capabilities**

2251 This section of air protocol specific parameters corresponds to LLRP parameters
2252 specified in section 9.

2253 **15.2.1.1.1 C1G2LLRPCapabilities Parameter**

2254 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2255 parameter. Readers MAY support BlockErase, and MAY support BlockWrite. Readers
2256 SHALL support at least one select filter per query.

C1G2LLRPCapabilities Parameter

CanSupportBlockErase: Boolean

CanSupportBlockWrite: Boolean

MaxNumSelectFiltersPerQuery: Unsigned Short Integer. If set to zero, it indicates there is no maximum limit.

2257 **15.2.1.1.2 UHFC1G2RFModeTable Parameter**

2258 This parameter carries the set of C1G2 RF modes that the Reader is capable of operating.

2259 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2260 parameter.

UHFC1G2RFModeTable Parameter

UHFC1G2RFModeSet: List of <UHFC1G2RFModeTableEntry Parameter>

2261 15.2.1.1.2.1 UHFC1G2RFModeTableEntry Parameter

2262 This parameter carries the information for each UHFC1G2 RF mode. A mode that has
2263 been tested for conformance by the EPCGlobal Hardware Action Group's Testing and
2264 Conformance (HAG T&C) group, is indicated using a conformance flag.

2265 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2266 parameter.

UHFC1G2RFModeTableEntry Parameter

Mode identifier: Unsigned Integer. This is a Reader defined identifier that the client may use to set the Gen2 operating parameters.

DR Value: Integer. Divide ratio.

Possible Values:

Value	DR
----	--
0	8
1	64/3

BDR Value: Integer. Backscatter data rate in bps.

Possible Values:

40000 – 640000 bps

M value: Integer. Modulation.

Possible Values:

Value	M
-----	---
0	FM0
1	2
2	4
3	8

Forward link modulation: Integer

Possible Values:

Value	Modulation
-----	-----
0	PR-ASK
1	SSB-ASK
2	DSB-ASK

PIE Value: Integer. One thousand times the ratio of data-0 symbol length and data-1 symbol length in pulse-interval encoding. The C1G2 spec specifies a ratio range of 1.5 – 2.0. (see section 6.3.1.2.4 in [C1G2]).

Possible Values:

1500-2000

MinTariValue: Integer. Minimum Tari time in nanoseconds (see section 6.3.1.2.4 in [C1G2])

Possible Values:

6250-25000

MaxTariValue: Integer. Maximum Tari time in nanoseconds. (see section 6.3.1.2.4 in [C1G2]).

Possible Values:

6250-25000

StepTariValue: Integer. Tari Step size in nanoseconds.(see section 6.3.1.2.4 in [C1G2])

Possible Values:

0 – 18750 nsec

Spectral Mask Indicator: Integer. Spectral mask characteristics of the mode. The Reader SHALL advertise this value if and only if the spectral mask value is valid for all the Tari steps in the range.

Possible Values:

Value	Modulation
-----	-----
0	Unknown
1	SI - Meets [C1G2] Single-Interrogator Mode Mask
2	MI - Meets [C1G2] Multi-Interrogator Mode Mask
3	DI - Meets [C1G2] Dense-Interrogator Mode Mask

EPC HAG T&C Conformance: Boolean. This flag indicates if the Reader vendor has received the certification for the parameter sets specified in this mode. The Reader SHALL set this flag to true only if the Reader vendor has received EPCGlobal conformance for this mode as specified in EPCGlobal Testing and Conformance.

2267 15.2.1.2 Inventory Operation

2268 This section of air protocol specific parameters corresponds to LLRP parameters
2269 specified in section 10.

2270 15.2.1.2.1 C1G2InventoryCommand Parameter

2271 This parameter defines the C1G2 inventory-specific settings to be used during a
2272 particular C1G2 inventory operation. This comprises of C1G2Filter Parameter, C1G2RF
2273 Parameter and C1G2Singulation Parameter. It is not necessary that the Filter, RF Control
2274 and Singulation Control Parameters be specified in each and every inventory command.
2275 They are optional parameters. If not specified, the default values in the Reader are used
2276 during the inventory operation. If multiple C1G2Filter parameters are encapsulated by the
2277 Client in the C1G2InventoryCommand parameter, the ordering of the filter parameters
2278 determine the order of C1G2 air-protocol commands (e.g., Select command) generated by
2279 the Reader.

2280 The TagInventoryStateAware flag is used to determine how to process all the C1G2Filter
 2281 and C1G2Singulation parameters in this command. At a functional level, if the Client is
 2282 managing the tag states during an inventory operation, it would set that flag to true and
 2283 pass the appropriate fields in the C1G2 Filter and C1G2 Singulation parameters. If a
 2284 reader set CanDoTagInventoryStateAwareSingulation to False in LLRPCapabilities
 2285 (section 9.2.2), it SHALL ignore the TagInventoryStateAware flag.

2286 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
 2287 parameter.

C1G2InventoryCommand Parameter TagInventoryStateAware: Boolean C1G2 Filter : List of <C1G2Filter Parameter> [if absent, use default] C1G2 RF: <C1G2RFControl Parameter> [if absent, use default] C1G2 Singulation Control : <C1G2SingulationControl Parameter> [if absent, use default] Custom Extension Point List: List of <Custom Parameter> (optional)
--

2288 **15.2.1.2.1.1 C1G2Filter Parameter**

2289 This parameter carries information specific to C1G2 filter (in particular, the parameters
 2290 for the select command) operation, and are optionally sent with each inventory command
 2291 from the Client to the Reader. This sets up the target tag population that gets inventoried.
 2292 For an inventory operation with multiple filters, multiple instances of filter parameters
 2293 are sent. A filter parameter contains the following fields:

- 2294 • Target tag mask: This contains the information for the tag memory data pattern
 2295 used for the select operation.
- 2296 • T: This value is set if the Client is interested in only a truncated portion of the tag
 2297 to be backscattered by the tag. The portion that gets backscattered includes the
 2298 portion of the tag ID following the mask. This bit has to be set only in the last
 2299 filter-spec.
- 2300 • TagInventoryStateAwareFilterAction: This is used if the TagInventoryStateAware
 2301 flag is set to true in the InventoryParameterSpec.
- 2302 • TagInventoryStateUnawareFilterAction: This is used if the
 2303 TagInventoryStateAware flag is set to false in the InventoryParameterSpec.

2304 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
 2305 parameter.

C1G2Filter Parameter Target Tag Mask: <C1G2TagInventoryMask Parameter> T: Integer <i>Possible Values:</i>
--

Value	Truncate action
-----	-----
0	Unspecified: The Reader decides what truncate action to take.
1	Do not truncate
2	Truncate

TagInventoryStateAwareAction: C1G2TagInventoryStateAwareFilterAction
Parameter (optional)

TagInventoryStateUnawareAction: C1G2TagInventoryStateUnawareFilterAction
Parameter (optional)

2306 15.2.1.2.1.1.1 C1G2TagInventoryMask Parameter

2307 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2308 parameter.

C1G2TagInventoryMask Parameter

MB: Integer. C1G2 Tag memory bank.

Possible Values:

1-3. The mask used for the C1G2 select command applies only to EPC, TID or User memory, and not to Reserved memory (MB 0).

Pointer: Unsigned Short Integer. The first (msb) bit location of the specified memory bank against which to compare the TagMask.

Length: Unsigned Short Integer. The number of bits to compare against the TagMask.

TagMask: Bit array. The pattern against which to compare.

2309 15.2.1.2.1.1.2 C1G2TagInventoryStateAwareFilterAction Parameter

2310 This parameter is used by the Client to manage the tag states during an inventory
2311 operation. In order to use this parameter during inventory, the TagInventoryStateAware
2312 flag is set to true in the InventoryParameterSpec. This parameter contains:

- 2313 • Target: This value indicates which flag in the tag to modify – whether the SL flag
2314 or its inventoried flag for a particular session.
- 2315 • Action describes the action for matching and non-matching tags. The actions are
2316 specific about the tag-inventory states - e.g., do nothing, assert or deassert SL,
2317 assign inventoried S0/S1/S2/S3 to A or B.

2318 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2319 parameter. Readers that do not support tag inventory state aware singulation SHALL set
2320 CanDoTagInventoryStateAwareSingulation to false in LLRPCapabilities.

C1G2TagInventoryStateAwareFilterAction Parameter

Target: Integer

Possible Values:

Value	Definition
-----	-----
0	SL
1	Inventoried state for session S0
2	Inventoried state for session S1
3	Inventoried state for session S2
4	Inventoried state for session S3

Action : Integer

Possible Values:

Value	Definition
-----	-----
0	Matching tags: assert SL or inventoried state $\rightarrow$ A. Non-matching tags: deassert SL or inventoried state $\rightarrow$ B.
1	Matching tags: assert SL or inventoried state $\rightarrow$ A. Non-matching tags: do nothing
2	Matching tags: do nothing Non-matching tags: deassert SL or inventoried state $\rightarrow$ B
3	Matching tags: negate SL or $(A \rightarrow B, B \rightarrow A)$ Non-matching tags: do nothing
4	Matching tags: deassert SL or inventoried state $\rightarrow$ B Non-matching tags: assert SL or inventoried state $\rightarrow$ A
5	Matching tags: deassert SL or inventoried state $\rightarrow$ B Non-matching tags: do nothing
6	Matching tags: do nothing Non-matching tags: assert SL or inventoried state $\rightarrow$ A
7	Matching tags: do nothing Non-matching tags: negate SL or $(A \rightarrow B, B \rightarrow A)$

2321 15.2.1.2.1.1.3 C1G2TagInventoryStateUnawareFilterAction Parameter

2322 This parameter is used by the Client if it does not want to manage the tag states during an
2323 inventory operation. Using this parameter, the Client instructs the Reader about the tags
2324 that should and should not participate in the inventory action. In order to use this
2325 parameter during inventory, the TagInventoryStateAware flag is set to false in the
2326 InventoryParameterSpec. This parameter contains:

- 2327 • Action describes the action for matching and non-matching tags. However, the
2328 action is simply specifying whether matching or non-matching tags partake in this
2329 inventory. The Reader is expected to handle the tag inventory states to facilitate
2330 this.

2331 In this parameter, Action=Select means search for pattern in Inventory, and
2332 Action=Unselect means do not search for pattern in Inventory.

2333 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2334 parameter.

C1G2TagInventoryStateUnawareFilterAction Parameter

Action : Integer

Possible Values:

Value -----	Matching Tags -----	Non-matching Tags -----
0	Select	Unselect
1	Select	Do nothing
2	Do nothing	Unselect
3	Unselect	Do nothing
4	Unselect	Select
5	Do nothing	Select

2335 15.2.1.2.1.2 C1G2RF Control Parameter

2336 This Parameter carries the settings relevant to RF forward and reverse link control in the
2337 C1G2 air protocol. This is basically the C1G2 RF Mode and the Tari value to use for the
2338 inventory operation.

2339 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2340 parameter.

C1G2RFControl Parameter

ModeIndex: Unsigned Integer. This is an index into the UHFC1G2RFModeTable.

Tari: Integer. Value of Tari to use for this mode specified in nsec. This is specified if the mode selected has a Tari range. If the selected mode has a range, and the Tari is set to zero, the Reader implementation picks up any Tari value within the range. If the selected mode has a range, and the specified Tari is out of that range and is not set to zero, an error message is generated.

Possible Values:

0 or 6250-25000 nsec

2341 15.2.1.2.1.3 C1G2SingulationControl Parameter

2342 This C1G2SingulationControl Parameter provides controls particular to the singulation
2343 process in the C1G2 air protocol. The singulation process is started using a Query
2344 command in the C1G2 protocol. The Query command describes the session number, tag
2345 state, the start Q value to use, and the RF link parameters. The RF link parameters are
2346 specified using the C1G2RFControl Parameter (see section 15.2.1.2.1.2). This
2347 Singulation Parameter specifies the session, tag state and description of the target
2348 singulation environment. The following attributes are specified to provide guidance to the
2349 Reader for the singulation algorithm:

- 2350 • Tag transit time: This is the measure of expected tag mobility in the field of view
2351 of the antenna where this inventory operation is getting executed.
- 2352 • Tag population: This is the expected tag population in the field of view of the
2353 antenna.

2354 In addition, the Singulation Parameter allows setting of the following:

- 2355 • Session ID: This is the C1G2 session number that the tags use to update the
2356 inventory state upon successful singulation.
- 2357 • TagInventoryStateAwareSingulationAction: This is used if the
2358 TagInventoryStateAware flag is set to true in the InventoryParameterSpec.
 - 2359 ○ I: This is the inventoried state of the target tag population in the selected
2360 session. Only tags that match the session state participate in the inventory
2361 round. If the Ignore value is specified, the Reader ignores this field, and
2362 its up to the Reader implementation to determine the value of I used in the
2363 inventory round.
 - 2364 ○ S: This is the state of the SL flag in the tag. Only tags that match that tag
2365 state participate in the inventory round. If the Ignore value is specified, the
2366 Reader ignores this field, and its up to the Reader implementation to
2367 determine the value of S used in the inventory round.
- 2368 If a reader sets CanDoTagInventoryStateAwareSingulation to False in LLRPCapabilities
2369 (section 9.2.2), it SHALL ignore the TagInventoryStateAwareSingulationAction field.
- 2370 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2371 parameter.

C1G2SingulationControl Parameter

Session: Integer. Session number to use for the inventory operation.

Possible Values:

0-3

Tag population: Unsigned Short Integer. An estimate of the tag population in view of the RF field of the antenna.

Tag transit time: Unsigned Integer. An estimate of the time a tag will typically remain in the RF field of the antenna specified in milliseconds.

TagInventoryStateAwareSingulationAction:

<C1G2TagInventoryStateAwareSingulationAction Parameter> (optional)

15.2.1.2.1.3.1 C1G2TagInventoryStateAwareSingulationAction Parameter

- 2372
- 2373 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2374 parameter. Readers that do not support tag inventory state aware singulation SHALL set
2375 CanDoTagInventoryStateAwareSingulation to false in LLRPCapabilities.

C1G2TagInventoryStateAwareSingulationAction Parameter

I : Integer

Possible Values:

Value	Definition
-----	-----
0	State A
1	State B

S : Integer

Possible Values:

Value	Definition
-----	-----
0	SL
1	~SL

15.2.1.3 Access Operation

This section of air protocol specific parameters corresponds to LLRP parameters specified in section 11.

15.2.1.3.1 C1G2TagSpec Parameter

This parameter describes the target tag population on which certain operations have to be performed. This Parameter is similar to the selection C1G2Filter Parameter described earlier. However, because these tags are stored in the Reader's memory and ternary comparisons are to be allowed for, each bit *i* in the target tag is represented using 2 bits - bit *i* in mask, and bit *i* in tag pattern. If bit *i* in the mask is zero, then bit *i* of the target tag is a don't care (X); if bit *i* in the mask is one, then bit *i* of the target tag is bit *i* of the tag pattern. For example, "all tags" is specified using a mask length of zero.

This parameter can carry up to two tag patterns. If more than one pattern is present, a Boolean AND is implied. Each tag pattern has a match or a non-match flag, allowing (A and B,!A and B, !A and !B, A and !B), where A and B are the tag patterns.

The tagSpec contains:

- TagPattern1
- TagPattern2

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

C1G2TagSpec Parameter

TagPattern1: <C1G2TargetTag Parameter>

TagPattern2: <C1G2TargetTag Parameter> [optional]

15.2.1.3.1.1 C1G2TargetTag Parameter

If Length is zero, this pattern will match all tags regardless of MB, pointer, mask and data.

Compliance requirement: Compliant Readers and Clients SHALL implement this parameter.

C1G2TargetTag Parameter

MB: Integer. Memory bank.

Possible Values: 0-3.

Pointer: Unsigned Short Integer. The address of the first (msb) bit against which to apply the Tag Mask and compare with the value.

Length: Unsigned Short Integer. The number of bits against which to apply the mask and compare the value.

TagMask : Bit array

TagData: Bit array

Match: Boolean

2400 **15.2.1.3.2 C1G2 OpSpec Parameters**

2401 This section describes the C1G2 specific OpSpec parameters that are sent as part of the
2402 AccessSpec. Each OpSpec parameter has an OpSpecID that is used when reporting
2403 results of the operation.

2404 **15.2.1.3.2.1 C1G2Read Parameter**

2405 MB is the memory bank to use. WordPtr is the starting word address. WordCount is the
2406 number of 16-bit words to be read. Following is text reproduced from the C1G2
2407 specification regarding WordCount=0. [If WordCount = 0, the tag backscatters the
2408 contents of the chosen memory bank starting at WordPtr and ending at the end of the
2409 bank, unless MB = 1, in which case the Tag shall backscatter the EPC memory contents
2410 starting at WordPtr and ending at the length of the EPC specified by the first 5 bits of the
2411 PC if WordPtr lies within the EPC, and shall backscatter the EPC memory contents
2412 starting at WordPtr and ending at the end of EPC memory if WordPtr lies outside the
2413 EPC.]

2414 Access Password is the password used by the Reader to transition the tag to the secure
2415 state so that it can read protected tag memory regions. For example, the Tag's Reserved
2416 memory is locked but not permalocked, meaning that the Reader must issue the access
2417 password and transition the Tag to the secured state before performing the read operation.

2418 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2419 parameter.

C1G2Read Parameter

OpSpecID: Unsigned Short Integer

MB: Integer. Memory bank.

Possible Values: 0-3

WordPtr: Unsigned Short Integer. The word addresss of the first word to read from the chosen memory bank.

WordCount: Unsigned Short Integer

AccessPassword: Unsigned Integer

2420 **15.2.1.3.2.2 C1G2Write Parameter**
2421 MB is the memory bank to use. WordPtr is the starting word address. Write Data is the
2422 data to be written to the tag. Word Count is the number of words to be written.
2423 Depending on the word count, the Reader may have to execute multiple C1G2 air
2424 protocol Write commands. Access Password is the password used by the Reader to
2425 transition the tag to the secure state so that it can write to protected tag memory regions.

2426 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2427 parameter.

C1G2Write Parameter

OpSpecID : Unsigned Short Integer

MB: Integer. Memory bank.

Possible Values: 0-3

WordPtr: Unsigned Short Integer. The word addresss of the first word to be written to the chosen memory bank.

Write Data: Short array. The data to write to the chosen memory bank.

AccessPassword: Unsigned Integer

2428 **15.2.1.3.2.3 C1G2Kill Parameter**
2429 Kill Password is the value of the kill password to be used or set.

2430 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2431 parameter.

C1G2Kill Parameter

OpSpecID : Unsigned Short Integer

Kill Password: Unsigned Integer

2432 **15.2.1.3.2.4 C1G2Lock Parameter**
2433 This parameter contains the definition of the access privilege updates
2434 (read/write/permalock) to be performed in various locations of the memory. The five data
2435 fields for which we can define access control using the lock command are: Kill Password,
2436 Access Password, EPC memory, TID memory and User memory. The access privilege
2437 updates are expressed as a list of C1G2LockPayload Parameters, one for each memory
2438 location.

2439 The Access Password provides the password to enter the *secured* state. A Reader can
2440 perform a lock operation on a tag only if the tag is in the *secured* state. The tag enters the
2441 secured state only using the Access Password (if a non-zero value).

2442 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2443 parameter.

C1G2Lock Parameter

OpSpecID : Unsigned Short Integer

LockCommandPayloadList: List of <C1G2LockPayload Parameter>

Access Password: Unsigned Integer

2444 15.2.1.3.2.4.1 C1G2LockPayload Parameter

2445 This parameter contains the definition of the access privilege updates
2446 (read/write/permalock) to be performed for a single location of the tag memory. The five
2447 data fields for which we can define access control using the lock command are: Kill
2448 Password, Access Password, EPC memory, TID memory and User memory.

2449 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2450 parameter.

C1G2LockPayload Parameter

OpSpecID : Unsigned Short Integer

Privilege: Integer. Value indicates the access privilege to be applied.

Possible Values:

Value	Access Privilege
-----	-----
0	Read/Write
1	Permalock
2	Permaunlock
3	Unlock

DataField: Unsigned Integer. Value indicates to which data field the access privilege will be applied.

Possible Values:

Value	Field
-----	-----
0	Kill Password
1	Access Password
2	EPC Memory
3	TID Memory
4	User Memory

2451 15.2.1.3.2.5 C1G2BlockErase Parameter

2452 MB is the memory bank to use. WordPtr is the starting word address. Word Count is the
2453 number of 16-bit words to be read. Access Password is the password used by the Reader
2454 to transition the tag to the secure state so that it can erase protected tag memory regions.

2455 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2456 parameter. Readers that do not support C1G2BlockErase SHALL set
2457 CanSupportBlockErase to false in C1G2LLRPCapabilities. If such a Reader receives an
2458 ADD_ACCESSSPEC with an AccessSpec that contained this OpSpec parameter, the
2459 Reader SHALL return an error for that message and not add the AccessSpec.

C1G2BlockErase Parameter

OpSpecID : Unsigned Short Integer

MB: Integer. Memory bank.

Possible Values: 0-3

WordPtr: Unsigned Short Integer. Word address of first word to be erased.

Word Count: Unsigned Short Integer. Number of words to erase.

Access Password: Unsigned Integer

2460 **15.2.1.3.2.6 C1G2BlockWrite Parameter**

2461 MB is the memory bank to use. WordPtr is the starting word address. Word Count is the
2462 number of 16-bit words to be written. Depending on the word count, the Reader may
2463 have to execute multiple C1G2 air protocol block write commands. Write Data is the data
2464 to be written to the tag. Access Password is the password used by the Reader to transition
2465 the tag to the secure state so that it can write to protected tag memory regions.

2466 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2467 parameter. Readers that do not support C1G2BlockWrite SHALL set
2468 CanSupportBlockWrite to false in C1G2LLRPCapabilities. If such a Reader receives an
2469 ADD_ACCESSSPEC with an AccessSpec that contained this OpSpec parameter, the
2470 Reader SHALL return an error for that message and not add the AccessSpec.

C1G2BlockWrite Parameter

OpSpecID : Unsigned Short Integer

MB: Integer. Memory bank.

Possible Values: 0-3

WordPtr: Unsigned Short Integer. Word address of first word to be written.

Write Data: Short array

Access Password: Unsigned Integer

2471 **15.2.1.4 Reader Device Configuration**

2472 This section of air protocol specific parameters corresponds to LLRP parameters
2473 specified in section 12. The only air protocol specific parameter is
2474 AirProtocolInventoryCommandSettings parameter in the AntennaConfiguration (section
2475 12.2.6). The C1G2 specific InventoryCommand is already defined in section 15.2.1.2.1.

2476 **15.2.1.5 Reports**

2477 This section of air protocol specific parameters corresponds to LLRP parameters
2478 specified in section 13.2.1.1.

2479 **15.2.1.5.1 C1G2EPCMemorySelector Parameter**

2480 This parameter is used to determine what contents are of interest in the C1G2EPC
2481 memory bank for reporting. If enableCRC and enablePC is set to false, only the EPC is
2482 returned in the RO Report. If enablePC is set to true, the PC bits and the EPC are returned
2483 in the RO Report. If enablePC and enableCRC is set to true, the EPC, PC bits and CRC
2484 are returned in the RO Report.

2485 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2486 parameter.

C1G2EPCMemorySelector

enablePC: Boolean

enableCRC: Boolean

2487 **15.2.1.5.2 C1G2PC Parameter**

2488 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2489 parameter.

C1G2PC Parameter

PC bits: Unsigned Short Integer

2490 **15.2.1.5.3 C1G2CRC Parameter**

2491 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2492 parameter.

C1G2CRC Parameter

CRC: Unsigned Short Integer

2493 **15.2.1.5.4 C1G2SingulationDetails Parameter**

2494 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2495 parameter.

C1G2SingulationDetails Parameter

NumCollisionSlots: Unsigned Short Integer. The number of slots detected as collided over the duration of this report.

NumEmptySlots: Unsigned Short Integer. The number of slots detected as empty over the duration of this report.

2496 **15.2.1.5.5 C1G2 OpSpec Results**

2497 **15.2.1.5.5.1 C1G2ReadOpSpecResult Parameter**

2498 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2499 parameter.

C1G2ReadOpSpecResult Parameter

OpSpecID: Unsigned Short Integer

ReadData: Short Array. The data read from the RFID tag.

Result: Integer

Possible Values:

Value	Definition
-----	-----
0	Success
1	Non-specific tag error
2	No response from tag
3	Non-specific reader error

2500 15.2.1.5.5.2 C1G2WriteOpSpecResult Parameter

2501 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2502 parameter.

C1G2WriteOpSpecResult Parameter

OpSpecID: Unsigned Short Integer

NumWordsWritten: Unsigned Short Integer. The number of words written as a result of this OpSpec. If the number of words written is not equal to the length of the data pattern to write, the Result below SHALL be non-zero.

Result: Integer

Possible Values:

Value	Definition
-----	-----
0	Success
1	Tag memory overrun error
2	Tag memory locked error
3	Insufficient power to perform memory-write operation
4	Non-specific tag error
5	No response from tag
6	Non-specific reader error

2503 15.2.1.5.5.3 C1G2KillOpSpecResult Parameter

2504 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2505 parameter.

C1G2KillOpSpecResult Parameter

OpSpecID: Unsigned Short Integer

Result: Integer

Possible Values:

Value	Definition
-----	-----

0	Success
1	Zero kill password error
2	Insufficient power to perform kill operation
3	Non-specific tag error
4	No response from tag
5	Non-specific reader error

2506 **15.2.1.5.5.4 C1G2LockOpSpecResult Parameter**

2507 **Compliance requirement:** Compliant Readers and Clients SHALL implement this
2508 parameter.

C1G2LockOpSpecResult Parameter

OpSpecID: Unsigned Short Integer

Result: Integer

Possible Values:

Value	Definition
-----	-----
0	Success
1	Insufficient power to perform lock operation
2	Non-specific tag error
3	No response from tag
4	Non-specific reader error

2509 **15.2.1.5.5.5 C1G2BlockEraseOpSpecResult Parameter**

2510 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2511 parameter. Readers that do not support C1G2 Block Erase SHALL set
2512 CanSupportBlockErase to false in C1G2LLRPCapabilities. If such a Reader receives an
2513 ADD_ACCESSSPEC with an AccessSpec that contains this OpSpec parameter, the
2514 Reader SHALL return an error for that message and not add the AccessSpec.

C1G2BlockEraseOpSpecResult Parameter

OpSpecID: Unsigned Short Integer

Result: Integer

Possible Values:

Value	Definition
-----	-----
0	Success
1	Tag memory overrun error
2	Tag memory locked error
3	Insufficient power to perform block erase operation
4	Non-specific tag error
5	No response from tag
6	Non-specific reader error

2515 **15.2.1.5.5.6 C1G2BlockWriteOpSpecResult Parameter**

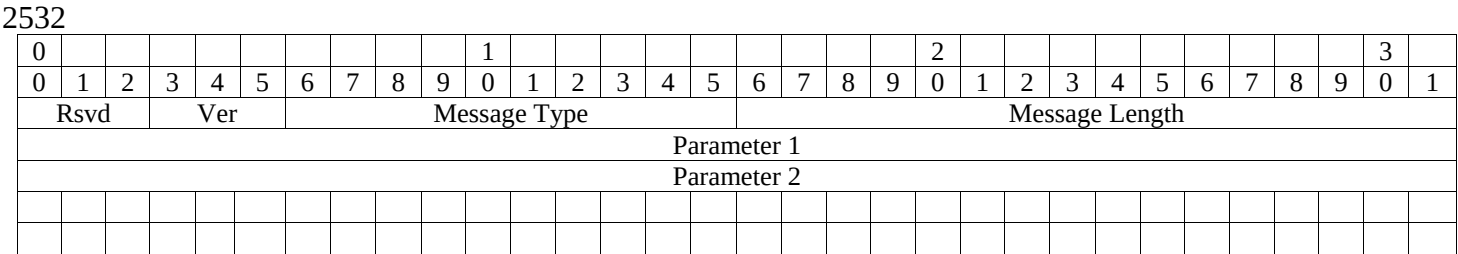
2516 **Compliance requirement:** Compliant Readers and Clients MAY implement this
2517 parameter. Readers that do not support C1G2 Block Write SHALL set
2518 CanSupportBlockWrite to false in C1G2LLRPCapabilities. If such a Reader receives an
2519 ADD_ACCESSSPEC with an AccessSpec that contains this OpSpec parameter, the
2520 Reader SHALL return an error for that message and not add the AccessSpec.

C1G2BlockWriteOpSpecResult Parameter	
OpSpecID: Unsigned Short Integer	
NumWordsWritten: Unsigned Short Integer	
Result: Integer	
<i>Possible Values:</i>	
Value	Definition
-----	-----
0	Success
1	Tag memory overrun error
2	Tag memory locked error
3	Insufficient power to perform memory-write operation
4	Non-specific tag error
5	No response from tag
6	Non-specific reader error

2521 **16 Binary Encoding for LLRP**

2522 This section contains the specific formats and operations for the binary encoding of the
2523 Low Level Reader Protocol. All fields and parameters must be encoded in the order
2524 shown in the diagrams in this section. This section does not contain information that has
2525 been generalized in the main body of the document. Refer to sections 8-15 for the
2526 description of the messages and the parameters and fields in the messages.

2527 The binary encoding is based on a stream of octets. Each octet represents 8 bits of
2528 information. Octets within the data stream are serialized according to the particular
2529 transport mechanism over which this binding is carried. Octet numbering shown in this
2530 section is in network order. For example, in Figure 12, the first octet that a LLRP
2531 endpoint receives contains Rsvd, Ver and 2 bits of Message type.



2533 **Figure 12: Network Order**

2534 Figure 13 illustrates the bit order inside the fields.

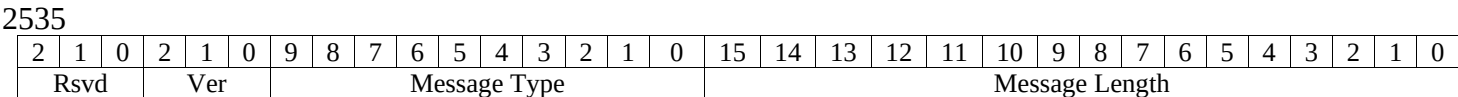


Figure 13: Bit Order in Fields

Integer numbers SHALL be encoded in network byte order with the most significant byte of the integer being sent first (big-Endian). Bit arrays are aligned to the most significant bit. Bit arrays are padded to an octet boundary. Pad bits SHALL be ignored by the Reader and Client.

The length of all messages within the binary encoding SHALL be multiples of octets. This means all parameters within the binary encoding SHALL be multiples of octets. This includes any custom or vendor specific parameter. All the messages and parameters in this section have been padded with zero to ensure that the length is a multiple of octets.

Notations

Inside a message or a parameter,

- If a parameter X is denoted simply as X, it means that X is mandatory and appears once in the message.
- If a parameter X is denoted as X (0-n), it means that X is optional in the message, and it can appear multiple times in the message.
- If a parameter X is denoted as X (0-1), it means that X is optional in the message and that it can appear at most once in the message.
- If a parameter X is denoted as X (1-n), it means that X is mandatory and can appear multiple times in the message.

Negative Numbers

Negative numbers are represented using twos complement notation.

16.1 Messages

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd			Ver			Message Type										Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																															
Message Value																															

- Reserved bits: 3 bits
The reserved bits are reserved for future extensions. All reserved bits in messages SHALL be set to 0 in outgoing messages.
- Ver: 3 bits
The version of LLRP. Implementations of LLRP based on this specification are using the value 0x1. Other values are reserved for future use.
- Message Type: 10 bits

2569 The type of LLRP message being carried in the message.
2570
2571 Message Length: 32 bits
2572 This value represents the size of the entire message in octets starting from bit offset 0 of the first word.
2573 Therefore, if the Message Value field is zero-length, the Length field will be set to 10.
2574
2575 Message ID: 32 bits
2576 As stated earlier, the communications between the Client and the Reader are primarily of a request-
2577 response type - requests/commands from the Client to the Reader, and responses from the Reader to the
2578 Client. In order to facilitate multiple outstanding commands/requests from the Client, LLRP uses a
2579 Message sequence number in each message. The Message sequence number is used to correlate a
2580 response with the original request. This sequence number is local to the LLRP channel.
2581
2582 Message Value: variable length
2583 Dependent on the Message Type.

2599 16.1.4 ADD_ROSPEC_RESPONSE

2600

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd			Ver			Message Type = 30										Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																															
LLRPStatus Parameter																															

2601

2602 See section 10.1.2.

2603

2604 16.1.5 DELETE_ROSPEC

2605

0										1									2									3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Rsvd			Ver			Message Type = 21										Message Length [31:16]																							
Message Length [15:0]																Message ID[31:16]																							
Message ID[15:0]																ROSpecID[31:16]																							
ROSpecID[15:0]																																							

2606

2607 See section 10.1.3.

2608

2609 16.1.6 DELETE_ROSPEC_RESPONSE

2610

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd			Ver			Message Type =31										Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																															
LLRPStatus Parameter																															

2611

2612 See section 10.1.4.

2613

2614 16.1.7 START_ROSPEC

2615

15															2															3														
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1													
Rsvd			Ver			Message Type = 22										Message Length [31:16]																												
Message Length [15:0]															Message ID[31:16]																													
Message ID[15:0]															ROSpecID[31:16]																													
ROSpecID[15:0]																																												

2616
2617 See section 10.1.5.
2618

2619 16.1.8 START_ROSPEC_RESPONSE

2620

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd		Ver		Message Type = 32												Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																LLRPStatus Parameter															

2621
2622 See section 10.1.6.
2623

2624 16.1.9 STOP_ROSPEC

0										1									2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1									
Rsvd		Ver		Message Type = 23												Message Length [31:16]																								
Message Length [15:0]																Message ID[31:16]																								
Message ID[15:0]																ROSpecID[31:16]																								
ROSpecID[15:0]																																								

2625
2626 See section 10.1.7.
2627

2628 16.1.10 STOP_ROSPEC_RESPONSE

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd		Ver		Message Type = 33												Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																															
LLRPStatus Parameter																															

2629
2630 See section 10.1.8.
2631

2632 16.1.11 ENABLE_ROSPEC

0										1									2										3										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Rsvd			Ver			Message Type = 24										Message Length [31:16]																							
Message Length [15:0]																Message ID[31:16]																							
Message ID[15:0]																ROSpecID[31:16]																							
ROSpecID[15:0]																																							

2668 16.1.19 DELETE_ACCESSSPEC

2669

0										1										2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1										
Rsvd		Ver		Message Type = 41												Message Length [31:16]																									
Message Length [15:0]																Message ID[31:16]																									
Message ID[15:0]																AccessSpecId[31:16]																									
AccessSpecId[15:0]																																									

2670

2671 See Section 11.1.3.

2672

2673 16.1.20 DELETE_ACCESSSPEC_RESPONSE

2674

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd		Ver		Message Type = 51												Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																LLRPStatus Parameter															

2675

2676 See Section 11.1.4.

2677

2678 16.1.21 ENABLE_ACCESSSPEC

2679

0										1										2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1										
Rsvd		Ver		Message Type = 42												Message Length [31:16]																									
Message Length [15:0]																Message ID[31:16]																									
Message ID[15:0]																AccessSpecId[31:16]																									
AccessSpecId[15:0]																																									

2680

2681 See Section 11.1.5.

2682

2683 16.1.22 ENABLE_ACCESSSPEC_RESPONSE

2684

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd			Ver			Message Type = 52										Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																LLRPStatus Parameter															

2685
2686 See Section 11.1.6.
2687

2688 16.1.23 DISABLE_ACCESSSPEC

2689

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Rsvd			Ver			Message Type = 43										Message Length [31:16]																							
Message Length [15:0]																Message ID[31:16]																							
Message ID[15:0]																AccessSpecId[31:16]																							
AccessSpecId[15:0]																																							

2690
2691 See Section 11.1.7.
2692

2693 16.1.24 DISABLE_ACCESSSPEC_RESPONSE

2694

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Rsvd			Ver			Message Type = 53										Message Length [31:16]																							
Message Length [15:0]																Message ID[31:16]																							
Message ID[15:0]																																							
LLRPStatus Parameter																																							

2695
2696 See Section 11.1.8.
2697

2698 16.1.25 GET_ACCESSSPECS

2699

0										1									2										3										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Rsvd			Ver			Message Type = 44										Message Length [31:16]																							
Message Length [15:0]																Message ID[31:16]																							
Message ID[15:0]																																							

2700
2701 See Section 11.1.9.
2702

2703 16.1.26 GET_ACCESSSPECS_RESPONSE

2704

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd			Ver			Message Type = 54										Message Length [31:16]															

Message Length [15:0]															Message ID[31:16]																																	
Message ID[15:0]																																																
LLRPStatus Parameter																																																
AccessSpec Parameter (0-n)																																																

2705
2706 See Section 11.1.10.
2707

2708 16.1.27 CLIENT_REQUEST_OP

2709

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd		Ver		Message Type = 45												Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																TagReportData Parameter															

2710
2711 See Section 11.1.11.

2712 16.1.28 CLIENT_REQUEST_OP_RESPONSE

2713

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd		Ver		Message Type = 55												Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																ClientRequestResponse Parameter															

2714
2715 See Section 11.1.12.
2716

2717 16.1.29 GET_REPORT

2718

0											1									2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1										
Rsvd		Ver		Message Type = 60												Message Length [31:16]																									
Message Length [15:0]																Message ID[31:16]																									
Message ID[15:0]																																									

2719
2720 See Section 13.1.1.
2721

2722 16.1.30 RO_ACCESS_REPORT

2723

0										1										2										3																	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																
Rsvd		Ver		Message Type = 61												Message Length [31:16]																															
Message Length [15:0]																Message ID[31:16]																															
Message ID[15:0]																																															
TagReportData Parameter (0-n)																																															
RFSurveyReportReportData Parameter (0-n)																																															
Custom Parameter (0-n)																																															

2724
2725 See Section 13.1.2.
2726

2727 16.1.31 KEEPALIVE

2728

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Rsvd			Ver			Message Type = 62										Message Length [31:16]																							
Message Length [15:0]																Message ID[31:16]																							
Message ID[15:0]																																							

2729
2730 See Section 13.1.3.
2731

2732 16.1.32 KEEPALIVE_ACK

2733

0										1										2										3										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1									
Rsvd			Ver			Message Type = 72										Message Length [31:16]																								
Message Length [15:0]																Message ID[31:16]																								
Message ID[15:0]																																								

2734
2735 See Section 13.1.4.

2736 16.1.33 READER_EVENT_NOTIFICATION

2737

0										1										2										3																	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																
Rsvd		Ver		Message Type = 63												Message Length [31:16]																															
Message Length [15:0]																Message ID[31:16]																															
Message ID[15:0]																ReaderEventNotificationData Parameter																															

2738

2739 See Section 13.1.5.

2740

2741 16.1.34 ENABLE_EVENTS_AND_REPORTS

2742

0										1										2									3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1							
Rsvd			Ver			Message Type = 64										Message Length [31:16]																						
Message Length [15:0]															Message ID[31:16]																							
Message ID[15:0]																																						

2743

2744 See Section 13.1.6.

2745 16.1.35 ERROR_MESSAGE

2746

0										1										2										3																	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																
Rsvd		Ver		Message Type = 100												Message Length [31:16]																															
Message Length [15:0]																Message ID[31:16]																															
Message ID[15:0]																LLRPStatus Parameter																															

2747

2748 See Section 14.1.1.

2749

2750 16.1.36 GET_READER_CONFIG

2751

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd			Ver			Message Type = 2										Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															

Message ID[15:0]															Antenna ID																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
RequestedData					GPIPortNum															GPOPortNum[15:8]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
GPOPortNum[7:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
Custom Parameter (0-n)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															

2752

2753 See Section 12.1.1.

2754

2755 16.1.37 GET_READER_CONFIG_RESPONSE

2756

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd		Ver		Message Type = 12												Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																															
LLRPStatus Parameter																															
Identification Parameter (0-1)																															
AntennaProperties Parameter (0-n)																															
AntennaConfiguration Parameter (0-n)																															
ReaderEventNotificationSpec Parameter (0-1)																															
ROReportSpec Parameter (0-1)																															
AccessReportSpec Parameter (0-1)																															
LLRPConfigurationStateValue Parameter (0-1)																															
KeepaliveSpec Parameter (0-1)																															
GPIPortCurrentState Parameter (0-n)																															
GPOWriteData Parameter (0-n)																															
EventsAndReports Parameter (0-1)																															
Custom Parameter (0-n)																															

2757

2758 See Section 12.1.2.

2759

2760 16.1.38 SET_READER_CONFIG

2761

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Rsvd			Ver			Message Type = 3										Message Length [31:16]															
Message Length [15:0]																Message ID[31:16]															
Message ID[15:0]																R	Reserved														
ReaderEventNotificationSpec Parameter (0-1)																															
AntennaProperties Parameter (0-n)																															
AntennaConfiguration Parameter (0-n)																															
ROResponseSpec Parameter (0-1)																															
AccessReportSpec Parameter (0-1)																															
KeepaliveSpec Parameter (0-1)																															
GPOWriteData Parameter (0-n)																															
GPIPortCurrentState Parameter (0-n)																															
EventsAndReports Parameter (0-1)																															
Custom Parameter (0-n)																															

2783 **16.1.42 CUSTOM_MESSAGE**

2784

0										1										2										3					
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
Rsvd			Ver			Message Type = 1023										Message Length [31:16]																			
Message Length [15:0]																Message ID[31:16]																			
Message ID[15:0]																Vendor Identifier [31:16]																			
Vendor Identifier [15:0]																Message Subtype																			
Vendor Specified Payload																																			

2785

2786 See Section 8.1.

2787

2788 **16.2 LLRP Parameters**

2789 LLRP parameters are defined in the following subsections with the exception that the air
2790 protocol specific LLRP parameters are defined in Section 16.3. The binary encoding of
2791 LLRP uses two different encodings of parameters: Type-length-value (TLV) encoded
2792 parameters, and Type-value (TV) encoded parameters. The TV encoding is only used for
2793 encoding parameters that are fixed-length, and are in Reports and Notifications from the
2794 Reader. The use of a compact encoding (i.e., TV) for the Reports and Notifications helps
2795 improve the network efficiency.

2796 **16.2.1 TLV and TV Encoding of LLRP Parameter**

2797 The type of encoding (TLV or TV) is determined based on the value of bit 0 in the
2798 parameter header. All TLV-encoded Parameters SHALL have a 0 in bit 0 of the header.
2799 All TV-encoded Parameters SHALL have a 1 in bit 0 of the header.

2800 **16.2.1.1 TLV-Parameters**

2801 LLRP TLV-Parameters have the following encoding structure.

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Parameter Type										Parameter Length															
Parameter Value																															

2802

2803

2804 Reserved bits: 6 bits

2805 The reserved bits are reserved for future extensions. All reserved bits SHALL be set to 0.

2806

2807 Parameter Type: 10 bits

2808 This is the type of LLRP parameter being carried in the message. The parameter number space for the

1	Parameter Type	Parameter Value																											

2848

2849 Parameter Type: 8 bits

2850 This is the type of LLRP parameter being carried in the message. The parameter number space for the
2851 TV-parameters is 1 – 127. The number space 128-2047 is reserved for TLV-parameters.

2852

2853 Parameter Value: variable length

2854 Dependent on the Parameter Type.

2855

2856 16.2.1.2.1 Encoding Guidelines for TV-Parameters

2857 The following rule applies to TV-Parameters:

- 2858 • TV-Parameters cannot contain sub-parameters (TLV or TV-Parameters).

2859

2860 16.2.2 General Parameters

2861 16.2.2.1 UTCTimestamp Parameter

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 128										Length															
Microseconds [63:32]																															
Microseconds [31:0]																															

2862

2863 See Section 7.1.3.1.1.1.

2864

2865 16.2.2.2 Uptime Parameter

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 129										Length															
MicroSeconds [63:32]																															
Microseconds [31:0]																															

2866

2867 See section 7.1.3.1.1.2.

2868

2869

2870 16.2.3 Reader Device Capabilities Parameters

2871 16.2.3.1 GeneralDeviceCapabilities Parameter

2872

0										1									2										3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Reserved						Type = 137										Length																	
MaxNumberOfAntennaSupported																C	T	Reserved															
Device manufacturer name																																	
Model Name																																	
FirmwareVersionByteCount																																	
Reader Firmware Version: Variable length UTF-8 String																																	
ReceiveSensitivityTableEntry Parameter (1-n)																																	
PerAntennaReceiveSensitivityRange Parameter (0-n)																																	
GPIOCapabilities Parameter																																	
PerAntennaAirProtocol Parameter (1-n)																																	

2873

2874 See Section 9.2.1.

2875

2876 **Abbreviations**

2877 C – CanSetAntennaProperties

2878 T - HasUTCClockCapability

2879

2880 **16.2.3.1.1 ReceiveSensitivityTableEntry Parameter**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 139										Length															
Index																Receive Sensitivity Value															

2881

2882 See Section 9.2.1.11.

2883

2884 **16.2.3.1.2 PerAntennaReceiveSensitivityRange Parameter**

0										1										2										3										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1									
Reserved						Type = 149										Length																								
AntennaID																ReceiveSensitivityIndexMin																								
ReceiveSensitivityIndexMax																																								

2885

2886 See Section 9.2.1.22.

2887 **16.2.3.1.3 PerAntennaAirProtocol Parameter**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 140										Length															
AntennaId												NumProtocols																			
ProtocolID#1								ProtocolID#2														ProtocolID#P									

2888

2889 See Section 9.2.1.33.

2890 **16.2.3.1.4 GPIOCapabilities Parameter**

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 141										Length															
NumGPIs										NumGPOs																					

2891

2892 See section 9.2.1.44.

2893

2894 **16.2.3.2 LLRPCapabilities Parameter**

0										1									2								3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 142										Length															
C	R	S	T	H	Reserved			MaxPriorityLevelSupported							ClientRequestOpSpecTimeout																
MaxNumROSpecs																															
MaxNumSpecsPerROSpec																															
MaxNumInventoryParameterSpecsPerAISpec																															
MaxNumAccessSpecs																															
MaxNumOpSpecsPerAccessSpec																															

2895

2896 **Abbreviations**

2897 C – CanDoRFSurvey

2898 R – CanReportBufferFillWarning

2899 S – SupportsClientRequestOpSpec

2900 T – CanDoTagInventoryStateAwareSingulation

2901 H – SupportsEventAndReportHolding

2902 MaxNumPriority – MaxNumPriorityLevelsSupported

2903

2904 See Section 9.2.2.

2905

2906 **16.2.3.3 AirProtocolLLRPCapabilities Parameter**

2907 See section 9.2.3.

2908

2909 There is no separate binary encoding for AirProtocolLLRPCapabilities. Each Air protocol's capabilities are
 2910 expressed in a different LLRP Parameter. Refer to Section 16.3 for air protocol specific capability
 2911 parameters. For example, the C1G2LLRPCapabilities Parameter (Section 16.3.1.1.1) carries the C1G2 air
 2912 protocol capabilities.

2913

2914 **16.2.3.4 RegulatoryCapabilities Parameter**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 143										Length															
Country Code																Communications Standard															
UHFBandCapabilities Parameter (0-1)																															

Custom Paramter (0-n)																																	

2915

2916 See Section 9.2.4.

2917

2918 16.2.3.4.1 UHFBandCapabilities Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 144										Length															
TransmitPowerLevelTableEntry Parameter (1-n)																															
FrequencyInformation Parameter																															
UHFC1G2RFModeTableEntry Parameter (1-n)																															

2919

2920 See Section 9.2.4.1.

2921

2922 16.2.3.4.1.1 TransmitPowerLevelTableEntry Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 145										Length															
Index																TransmitPowerValue															

2923

2924 See Section 9.2.4.1.1.

2925

2926 16.2.3.4.1.2 FrequencyInformation Parameter

0										1									2											3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 146										Length															
H	Reserved					FrequencyHopTable Parameter (0-n)																									
FixedFrequencyTable (0-1)																															

2927

2928 Abbreviations

2929 H – Hopping

2930

2931 See Section 9.2.4.1.2.

2932

2933 16.2.3.4.1.2.1 FrequencyHopTable Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 147										Length															
HopTableId								Reserved								NumHops															
Frequency#1																															

....																																		
Frequency#n																																		

2934
2935 NumHops: Number of entries in the List of Frequencies.
2936
2937 See Section 9.2.4.1.2.1.

2938 16.2.3.4.1.2.2 FixedFrequencyTable Parameter

0										1										2											3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1							
Reserved						Type = 148										Length																						
NumFrequencies																Frequency#1[31:16]																						
Frequency#1[15:0]																																						
....																																						
Frequency#n [15:0]																																						

2939 NumFrequencies: Number of entries in the List of Frequencies.
2940
2941 See Section 9.2.4.1.2.2.
2942

2943 **16.2.4 Reader Operations Parameters**

2944 **16.2.4.1 ROSpec Parameter**

0										1									2											3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Reserved						Type = 177										Length																	
ROSpecID																																	
Priority										CurrentState																							
ROBoundarySpec Parameter																																	
SpecParameter (1-n) [See notes below]																																	
ROReportSpec Parameter (0-1)																																	

2945
2946 **Notes**
2947 Each SpecParameter can be one of the following types: AISpec Parameter or RFSurveySpec Parameter or
2948 Custom Parameter.
2949
2950 See Section 10.2.1.
2951

2952 **16.2.4.1.1 ROBoundarySpec Parameter**

0										1									2										3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Reserved						Type = 178										Length																	
ROSpecStartTrigger Parameter																																	
ROSpecStopTrigger Parameter																																	

2953
2954 See Section 10.2.1.1.
2955

2956 **16.2.4.1.1.1 ROSpecStartTrigger Parameter**

0											1									2										3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1						
Reserved						Type = 179										Length																					
ROSpecStartTriggerType																																					
PeriodicTriggerValue Parameter (0-1)																																					
GPITriggerValue Parameter (0-1)																																					

2957
2958 See Section 10.2.1.1.1.
2959

2960 **16.2.4.1.1.1.1 PeriodicTriggerValue Parameter**

0										1										2									3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 180										Length															
Offset																															
Period																															
UTCTimestamp Parameter (0-1)																															

2961
2962 See Section 10.2.1.1.1.1.
2963

2964 **16.2.4.1.1.1.2 GPITriggerValue Parameter**

0										1									2									3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Reserved						Type = 181										Length																							
GPIPortNum																E	Reserved						Timeout[31:24]																
Timeout [23:0]																																							

2965
2966 **Abbreviations**
2967 E – GPIEvent
2968
2969 See section 10.2.1.1.1.2.
2970

2971 **16.2.4.1.1.2 ROSpecStopTrigger Parameter**

0										1										2									3								
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1						
Reserved						Type = 182										Length																					
ROSpecStopTriggerType										DurationTriggerValue[31:8]																											
DurationTriggerValue[7:0]																																					
GPITriggerValue Parameter (0-1)																																					

2972
2973 See section 10.2.1.1.2.
2974

2975 16.2.4.2 AISpec Parameter

0										1									2										3						
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
Reserved						Type = 183										Length																			
										AntennaCount										AntennaID#1															
																				AntennaID#n															
AISpecStopTrigger Parameter																																			
InventoryParameter Spec Parameter (1-n)																																			
Custom Parameter (0-n)																																			

2976
2977 See section 10.2.2.
2978

2979 16.2.4.2.1 AISpecStopTrigger Parameter

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 184										Length															
AISpecStopTriggerType								DurationTrigger[31:24]																							
DurationTrigger[7:0]																															
GPITriggerValue Parameter (0-1)																															
TagObservationTrigger Parameter (0-1)																															

2980
2981 See section 10.2.2.1.
2982

2983 16.2.4.2.1.1 TagObservationTrigger Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 185										Length															
TriggerType									Reserved							NumberOfTags															
NumberOfAttempts																T															
Timeout																															

2984
2985 See section 10.2.2.1.1.
2986

2987 16.2.4.2.2 InventoryParameterSpec Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 186										Length															
InventoryParameterSpecID																ProtocolID															
AntennaConfigurationParameter (0-n)																															

Custom Parameter (0-n)																															

2988

2989 See section 10.2.2.2.

2990

2991 16.2.4.3 RFSurveySpec Parameter

0										1									2										3																								
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																						
Reserved						Type = 187										Length																																					
AntennaID																StartFrequency[31:16]																																					
StartFrequency[15:0]																EndFrequency[31:16]																																					
EndFrequency[15:0]																																																					
RFSurveySpecStopTrigger Parameter																																																					
Custom Parameter (0-n)																																																					

2992

2993 See Section 10.2.3.

2994

2995 16.2.4.3.1 RFSurveySpecStopTrigger Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 188										Length															
StopTriggerType						Duration [31:24]																									
Duration [7:0]						N[31:8]																									
N[7:0]																															

2996

2997 See Section 10.2.3.1.

2998

2999 16.2.5 Access Operation Parameters

3000 16.2.5.1 AccessSpec Parameter

0										1										2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
Reserved						Type = 207										Length																
AccessSpecID																																
AntennaId																ProtocolId								C	Reserved							
ROSpecID																																
AccessSpecStopTrigger Parameter																																
AccessCommand Parameter																																
AccessReportSpec Parameter (0-1)																																
Custom Parameter (0-n)																																

3001 Abbreviations

3002 C – CurrentState

3003

3004 See section 11.2.1.
3005

3006 16.2.5.1.1 AccessSpecStopTrigger Parameter

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 208										Length															
AccessSpecStopTrigger						OperationCountValue																									

3007
3008 See Section 11.2.1.1.
3009

3010 16.2.5.1.2 AccessCommand Parameter

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 209										Length															
TagSpecParameter [See notes below]																															
OpSpecParameter (1-n) [See notes below]																															
Custom Parameter (0-n)																															

3011
3012 **Notes**
3013 TagSpecParameter is the air protocol specific tag spec parameter. For C1G2, it is C1G2TagSpec Parameter.
3014
3015 Each OpSpecParameter can be one of two types: Air protocol specific OpSpec (e.g., C1G2OpSpec
3016 Parameter) or ClientRequestOpSpec Parameter.
3017
3018 See Section 11.2.1.2.

3019 16.2.5.1.3 ClientRequestOpSpec Parameter

0										1										2									3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1					
Reserved						Type = 210										Length																				
OpSpecID																																				

3020
3021 See Section 11.2.1.2.1.

3022 16.2.5.1.3.1 ClientRequestResponse Parameter

3023																															
0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 211										Length															
AccessSpecID																															
EPCDataParameter																															
OpSpecParameter (0-n) [See notes below]																															

3024

3025 **Notes**
 3026 Each OpSpecParameter is an Air protocol specific opspec (e.g., C1G2OpSpec Parameter).
 3027
 3028 See Section 11.2.2.

3029 16.2.6 Configuration Parameters

3030 16.2.6.1 LLRPConfigurationStateValue Parameter

3031

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 217										Length															
LLRPConfigurationStateValue																															

3032
 3033 See section 12.2.1.

3034 16.2.6.2 Identification Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 218										Length															
IDType								ByteCount																							
Reader ID(Variable length)																															

3035
 3036 See Section 12.2.2.

3037 16.2.6.3 GPOWriteData Parameter

0										1										2									3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 219										Length															
GPO Port Number																W	Reserved														

3038
 3039 **Abbreviations**
 3040 W – GPO Data
 3041
 3042 See Section 12.2.3.

3043 16.2.6.4 KeepaliveSpec Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 220										Length															
KeepaliveTriggerType								TimeInterval																							
TimeInterval																															

3044
3045 See Section 12.2.4.

3046 16.2.6.5 AntennaProperties Parameter

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 221										Length															
C	Reserved						AntennaId																AntennaGain[15:8]								
AntennaGain[7:0]																															

3047
3048 **Abbreviations**
3049 C – Antenna connected
3050
3051 See Section 12.2.5.
3052

3053 16.2.6.6 AntennaConfiguration Parameter

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 222										Length															
AntennaId																															
RFReceiver Parameter (0-1)																															
RFTransmitter Parameter (0-1)																															
AirProtocolInventoryCommandSettings Parameter (0-n)																															

3054
3055 **Notes:**
3056 Each AirProtocolInventoryCommandSettingsParameter instance is an Air protocol specific Parameter (e.g.,
3057 C1G2InventoryCommand Parameter).
3058
3059 See Section 12.2.6.
3060

3061 16.2.6.7 RFReceiver Parameter

0										1										2										3						
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1					
Reserved						Type = 223										Length																				
Receiver Sensitivity																																				

3062
3063 See Section 12.2.6.1.
3064

3065 16.2.6.8 RFTransmitter Parameter

0										1										2										3										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1									
Reserved						Type = 224										Length																								
HopTableId																ChannelIndex																								
TransmitPower																																								

[illegible]

3066

3067 See Section 12.2.6.2.

3068

3069 16.2.6.9 GPIPortCurrentState Parameter

[illegible]

3070

3071

3072 **Abbreviations**

3073 C – GPICConfig

3074

3075 See Section 12.2.6.3.

3076

3077 16.2.6.10 EventsAndReports Parameter

[illegible]

3078

3079 Abbreviations

3080 H – HoldEventsAndReportsUponReconnect

3081

3082 See Section 12.2.6.4

3083

3084 16.2.7 Reporting Parameters

3085 16.2.7.1 ROReportSpec Parameter

[illegible]

3086

3087 See Section 13.2.1.

3088

3089 16.2.7.1.1 *TagReportContentSelector* Parameter

[illegible]

OpSpecResultParameter (0-n) [See notes below]																													
Custom Parameter (0-n)																													

3113

3114 **Notes:**

3115 The EPDataParameter is either the EPData Parameter (Section 16.2.7.3.1) or EPC-96 Parameter
3116 (Section 16.2.7.3.2). The EPData Parameter SHALL be used for encoding a non-96 bit EPC, whereas the
3117 EPC-96 Parameter SHALL be used for encoding a 96-bit EPC.

3118

3119 The AirProtocolTagDataParameter is one or more air protocol specific tag data parameters (e.g., C1G2PC
3120 and C1G2CRC). In the C1G2 case, each parameter, C1G2PC and C1G2CRC, is optional in the
3121 TagReportData Parameter.

3122

3123 OpSpecResultParameter: Either an air protocol specific OpSpec result parameter (e.g., C1G2OpSpecResult
3124 Parameter) or ClientRequestOpSpecResult Parameter.

3125

3126 See section 13.2.3.

3127 **16.2.7.3.1 EPData Parameter**

0										1										2									3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 241										Length															
EPCLengthBits																															
EPC																															

3128

3129 EPCLengthBits: Number of bits in the EPC.

3130 See Section 13.2.3.2.

3131

3132 **16.2.7.3.2 EPC-96 Parameter (TV-Encoding)**

0										1										2									3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1					
1	Type=13								EPC[95:72]																											
EPC[71:40]																																				
EPC[39:8]																																				
EPC[7:0]																																				

3133

3134 See Section 13.2.3.2.

3135 **16.2.7.3.3 ROSpecID Parameter (TV-Encoding)**

0										1										2									3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=9								ROSpecID[31:8]																						
ROSpecID[7:0]																															

3136

3137 See Section 13.2.3.3.

3138

3139 **16.2.7.3.4 SpecIndex Parameter (TV-Encoding)**

0										1									2								3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=14								SpecIndex																						

3140

3141 See Section 13.2.3.4.

3142 **16.2.7.3.5 InventoryParameterSpecID Parameter (TV-Encoding)**

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=10								InventoryParameterSpecId																						

3143

3144 See Section 13.2.3.5.

3145 **16.2.7.3.6 AntennaID Parameter (TV-Encoding)**

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=1								AntennaId																						

3146

3147 See Section 13.2.3.6.

3148 **16.2.7.3.7 PeakRSSI Parameter (TV-Encoding)**

0										1									2									3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
1	Type=6								PeakRSSI																							

3149

3150 See Section 13.2.3.7.

3151 **16.2.7.3.8 ChannelIndex Parameter (TV-Encoding)**

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=7								ChannelIndex																						

3152

3153 See Section 13.2.3.8.

3154 **16.2.7.3.9 FirstSeenTimestampUTC Parameter (TV-Encoding)**

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=2								Microseconds [63:40]																						
Microseconds [39:8]																															
Microseconds[7:0]																															

3155

3156 See Section 13.2.3.9.

3157 **16.2.7.3.10 FirstSeenTimestampUptime Parameter (TV-Encoding)**

0										1									2									3	
---	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	---	--

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=3							Microseconds[63:40]																							
Microseconds [39:8]																															
Microseconds[7:0]																															

3158

3159 See Section 13.2.3.10.

3160 **16.2.7.3.11 LastSeenTimestampUTC Parameter (TV-Encoding)**

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=4							Microseconds[63:40]																							
Microseconds[39:8]																															
Microseconds[7:0]																															

3161

3162 See Section 13.2.3.11.

3163 **16.2.7.3.12 LastSeenTimestampUptime Parameter (TV-Encoding)**

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=5							Microseconds[63:40]																							
Microseconds[39:8]																															
Microseconds[7:0]																															

3164

3165 See Section 13.2.3.12.

3166 **16.2.7.3.13 TagSeenCount Parameter (TV-Encoding)**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=8							TagCount																							

3167

3168 See Section 13.2.3.13.

3169 **16.2.7.3.14 ClientRequestOpSpecResult Parameter (TV-Encoding)**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=15							OpSpecID																							

3170

3171 See Section 13.2.3.14.

3172 **16.2.7.3.15 AccessSpecID Parameter (TV-Encoding)**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=16							AccessSpecID[31:8]																							
AccessSpecID[7:0]																															

3173

3174 See Section 13.2.3.15.

3175

3176

3177 **16.2.7.4 RFSurveyReportData Parameter**

3178

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 242										Length															
ROSpecID Parameter (0-1)																															
SpecIndex Parameter (0-1)																															
FrequencyRSSIlevelEntry Parameter (1-n)																															
Custom Parameter (0-n)																															

3179

3180 See Section 13.2.3.15.

3181

3182 **16.2.7.4.1 FrequencyRSSIlevelEntry Parameter**

3183

0										1									2									3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1						
Reserved						Type = 243										Length																					
																		Frequency																			
																		Bandwidth																			
Average RSSI										Peak RSSI																											
TimestampParameter [See notes below]																																					

3184

3185 **Notes:**

3186 TimestampParameter: Either UTCTimestamp Parameter or UptimeParameter.

3187

3188 See section 13.2.4.1

3189

3190 **16.2.7.5 ReaderEventNotificationSpec Parameter**

3191

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 244										Length															
EventNotificationState Parameter(1-n)																															

3192

3193 See Section 13.2.5.

3194

3195

3196 16.2.7.5.1 EventNotificationState Parameter

3197

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 245										Length															
EventType																S	Reserved														

3198

3199 **Abbreviations:**

3200 S – NotificationState

3201

3202 See Section 13.2.5.1.

3203

3204 16.2.7.6 ReaderEventNotificationData Parameter

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved					Type = 246											Length															
TimestampParameter [See notes below]																															
HoppingEvent Parameter (0-1)																															
GPISpecEvent Parameter (0-1)																															
ROSpecEvent Parameter (0-1)																															
ReportBufferLevelWarningEvent Parameter (0-1)																															
ReportBufferOverflowErrorEvent Parameter (0-1)																															
ReaderExceptionEvent Parameter (0-1)																															
RFSurveyEvent Parameter (0-1)																															
AISpecEvent Parameter (0-1)																															
AntennaEvent Parameter (0-1)																															
ConnectionAttemptEvent Parameter (0-1)																															
ConnectionCloseEvent Parameter (0-1)																															
Custom Parameter (0-n)																															

3205

3206 **Notes:**

3207 TimestampParameter: Either UTCTimestamp Parameter or Uptime Parameter.

3208

3209 See section 13.2.6.

3210

3211 16.2.7.6.1 HoppingEvent Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 247										Length															
HopTableID																NextChannelIndex															

3212

3213 See section 13.2.6.2.

3214

3215 16.2.7.6.2 GPIEvent Parameter

0										1									2										3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Reserved						Type = 248										Length																	
GPIPortNumber																E	Reserved																

3216 Abbreviations

3217 E – GPIEvent

3218

3219 See section 13.2.6.3.

3220

3221 16.2.7.6.3 ROSpecEvent Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 249										Length															
EventType										ROSpecID[31:8]																					
ROSpecID[7:0]										PreemptingROSpecID[31:8]																					
PreemptingROSpecID[7:0]																															

3222

3223 See section 13.2.6.4.

3224

3225 16.2.7.6.4 ReportBufferLevelWarningEvent Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 250										Length															
ReportBufferPercentageFull																															

3226

3227 See section 13.2.6.5.

3228

3229 16.2.7.6.5 ReportBufferOverflowErrorEvent Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 251										Length															

3230

3231 See section 13.2.6.6.

3232

3233 16.2.7.6.6 ReaderExceptionEvent Parameter

										1																				2																				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																												
Reserved						Type = 252										Length																																											
Message String ByteCount																																																											

Message: Variable length UTF-8String																															
ROSpecID Parameter (0-1)																															
SpecIndex Parameter (0-1)																															
InventoryParameterSpecID Parameter (0-1)																															
AntennaID Parameter (0-1)																															
AccessSpecID Parameter (0-1)																															
OpSpecID Parameter (0-1)																															
Custom Parameter (0-n)																															

3234

3235 See Section 13.2.6.7.

3236

3237 **16.2.7.6.6.1 OpSpecID Parameter (TV-Encoding)**

0											1										2									3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=17										OpSpecID																				

3238

3239 See Section 13.2.5.7.1.

3240 **16.2.7.6.7 RFSurveyEvent Parameter**

0											1									2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 253										Length															
EventType										ROSpecID[31:8]																					
ROSpecID[7:0]										SpecIndex[15:0]																					

3241

3242 See Section 13.2.6.7.1.

3243

3244

3245 **16.2.7.6.8 AISpecEvent Parameter**

0											1									2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 254										Length															
EventType						ROSpecID[31:8]																									
ROSpecID[7:0]						SpecIndex[15:0]																									
AirProtocolSingulationDetailsParameter (0-1) [See notes below]																															

3246

3247 See section 13.2.6.9.

3248 **Notes:**

3249

3250 AirProtocolSingulationDetailsParameter is one of the air protocol specific singulation parameters (e.g.,

3251 C1G2SingulationDetails Parameter).

3252

3253 **16.2.7.6.9 AntennaEvent Parameter**

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 255										Length															
EventType						AntennaID																									

3254

3255 See Section 13.2.6.10.

3256

3257 **16.2.7.6.10 ConnectionAttemptEvent Parameter**

0										1									2										3					
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1			
Reserved						Type = 256										Length																		
Status																																		

3258

3259 See Section 13.2.6.11.

3260

3261 **16.2.7.6.11 ConnectionCloseEvent Parameter**

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 257										Length															

3262

3263 See Section 13.2.6.12.

3264

3265 **16.2.8 LLRP Error Parameters**

3266 **16.2.8.1 LLRPStatus Parameter**

3267

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 287										Length															
						ErrorCode										Error Description ByteCount															
Error Description: Variable length UTF-8 String																															
FieldError Parameter (0-1)																															
ParameterError Parameter (0-1)																															

3268

3269 See Section 14.2.2.

3270

3271 **16.2.8.1.1 FieldError Parameter**

3272

0										1									2										3	
---	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	---	--

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 288										Length															
FieldNum																ErrorCode															

3273

3274 See section 14.2.2.1.

3275

3276 **16.2.8.1.2 ParameterError Parameter**

3277

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 289										Length															
ParameterType																ErrorCode															
Field Error Parameter (0-1)																															
Parameter Error Parameter (0-1)																															

3278

3279 See Section 14.2.2.2.

3280

3281 **16.2.9 Custom Parameter**

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type=1023										Parameter Length															
Vendor ID																															
Subtype																															
VendorParameter Value																															

3282

3283 See Section 8.2.

3284

3285 **16.3 Air Protocol Specific Parameters**

3286 This section defines air protocol specific parameter encodings. There is a separate
3287 subsection here for each air protocol defined by LLRP. See section 15, in the LLRP
3288 abstract specification, for more information regarding air protocol specific parameters.

3289 **16.3.1 Class-1 Generation-2 (C1G2) Protocol Parameters**

3290 The Class-1 Generation-2 (C1G2) Air Protocol is specified by the EPCglobal Class-1
3291 Generation-2 UHF RFID Protocol v1.1.0 specification.

3292 The following subsections specify LLRP air protocol specific parameter encodings.
3293 These subsections are partitioned to correlate with subsections of section 16.2:

- 3294 - Capabilities Parameters
- 3295 - Reader Operations Parameters

- 3296 - Access Operation Parameters
- 3297 - Configuration Parameters
- 3298 - Reporting Parameters

3299 **16.3.1.1 Capabilities Parameters**

3300 This section of air protocol specific parameters corresponds to LLRP parameters
 3301 encodings specified in section 15.2.1.1.

3302 **16.3.1.1.1 C1G2LLRPCapabilities Parameter**

0											1									2									3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type =327										Length															
E	W	Reserved						MaxNumSelectFiltersPerQuery																							

3303
 3304 **Abbreviations**
 3305 E – CanSupportBlockErase
 3306 W – CanSupportBlockWrite
 3307
 3308 See Section 15.2.1.1.1.
 3309

3310 **16.3.1.1.2 UHFC1G2RFModeTable Parameter**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 328										Length															
UHFC1G2RFModeTableEntry Parameter (1-n)																															

3311
 3312 See Section 15.2.1.1.2.
 3313

3314 **16.3.1.1.2.1 UHFC1G2RFModeTableEntry Parameter**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 329										Length															
Mode identifier																															
R	C	Reserved						Mod						FLM						M											
BDR Value																															
PIE Value																															
MinTariValue																															
MaxTariValue																															
StepTariValue																															

3315
 3316 **Abbreviations**
 3317 R – DR Value

3318 M– Spectral Mask Indicator
 3319 Mod – M value / Modulation
 3320 FLM – Forward Link Modulation
 3321 C – EPC HAG T&C Conformance
 3322
 3323 See section 15.2.1.1.2.1.
 3324

3325 16.3.1.2 Reader Operations Parameters

3326 This section of air protocol specific parameters corresponds to LLRP parameters
 3327 encodings specified in section 15.2.1.2.

3328 16.3.1.2.1 C1G2InventoryCommand Parameter

0										1										2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
Reserved						Type = 330										Length																
S	Reserved								C1G2Filter Parameter (0-n) C1G2RFControl Paremeter (0-1) C1G2SingulationControl Parameter (0-1) Custom Parameter (0-n)																							

3329
 3330 **Abbreviations**
 3331 S – TagInventoryStateAware
 3332
 3333 See Section 15.2.1.2.1
 3334

3335 16.3.1.2.1.1 C1G2Filter Parameter

0										1									2										3																												
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																										
Reserved						Type =331										Length																																									
T	Reserved					C1G2TagInventoryMask Parameter																																																			
C1G2TagInventoryStateAwareFilterAction Paremeter (0-1)																																																									
C1G2TagInventoryStateUnawareFilterAction Parameter (0-1)																																																									

3336
 3337 See Section 15.2.1.2.1.1.
 3338

3339 16.3.1.2.1.1.1 C1G2TagInventoryMask Parameter

0											1									2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 332										Length															
MB	Reserved					Pointer[15:0]															MaskBitCount[15:8]										
MaskBitCount[7:0]																															
Tag Mask																															

3361

3362 16.3.1.2.1.3.1 C1G2TagInventoryStateAwareSingulationAction Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved					Type = 337										Length																
I	S	Reserved																													

3363

3364 See section 15.2.1.2.1.3.1.

3365

3366 16.3.1.3 Access Operation Parameters

3367 This section of air protocol specific parameters corresponds to LLRP parameters

3368 encodings specified in section 15.2.1.3.

3369 16.3.1.3.1 C1G2TagSpec Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 338										Length															
C1G2TargetTag Parameter																															
C1G2TargetTag Parameter (0-1)																															

3370

3371 See section 15.2.1.3.1.

3372

3373 16.3.1.3.1.1 C1G2TargetTag Parameter

3374

0										1									2										3								
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1						
Reserved						Type = 339										Length																					
MB	M	Resvd							Pointer													MaskBitCount[15:8]															
MaskBitCount[7:0]																																					
Tag Mask																																					
DataBitCount																																					
Tag Data																																					

3375

3376 Abbreviations

3377 M – Match.

3378

3379 See section 15.2.1.3.1.1.

3380

3381 16.3.1.3.2 C1G2 OpSpecs

3382 **16.3.1.3.2.1 C1G2Read Parameter**

3383

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 341										Length															
OpSpecID																AccessPassword[31:16]															
AccessPassword[15:0]																MB		Reserved						WordPointer[15:8]							
WordPointer[7:0]						WordCount																									

3384

3385 See section 15.2.1.3.2.2.

3386 **16.3.1.3.2.2 C1G2Write Parameter**

3387

0										1										2										3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Reserved						Type = 342										Length																	
OpSpecID																AccessPassword[31:16]																	
AccessPassword[15:0]																MB		Reserved						WordPointer[15:8]									
WordPointer[7:0]						WriteDataWordCount																											
Write Data																																	

3388 See section 15.2.1.3.2.2.

3389

3390 **16.3.1.3.2.3 C1G2Kill Parameter**

0										1									2										3								
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1						
Reserved						Type = 343										Length																					
OpSpecID																KillPassword[31:16]																					
KillPassword[15:0]																																					

3391

3392 See section 15.2.1.3.2.3.

3393 **16.3.1.3.2.4 C1G2Lock Parameter**

3394

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 344										Length															
OpSpecID																AccessPassword[31:16]															
AccessPassword[15:0]																															
C1G2LockPayload Parameter (1-n)																															

3395

3396 See section 15.2.1.3.2.4.
3397

3398 16.3.1.3.2.4.1 C1G2LockPayload Parameter

3399

0										1									2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 345										Length															
Privilege									DataField																						

3400

3401 See section 15.2.1.3.2.4.1.

3402

3403 16.3.1.3.2.5 C1G2BlockErase Parameter

3404

0										1									2										3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Reserved						Type = 346										Length																	
OpSpecID																AccessPassword[31:16]																	
AccessPassword[15:0]																MB		Reserved						WordPointer[15:8]									
WordPointer[7:0]						WordCount																											

3405

3406 See section 16.3.1.3.2.5.

3407

3408 16.3.1.3.2.6 C1G2BlockWrite Parameter

3409

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 347										Length															
OpSpecID																AccessPassword[31:16]															
AccessPassword[15:0]																MB		Reserved						WordPointer[15:8]							
WordPointer[7:0]						WriteDataWordCount																									
Write Data																															

3410

3411 See section 15.2.1.3.2.6.

3412

3413 16.3.1.4 Configuration Parameters

3414 This section of air protocol specific parameters corresponds to LLRP parameters
3415 specified in Section 12.2. The only air protocol specific parameter is the
3416 AirProtocolInventoryCommandSettings parameter in the AntennaConfiguration (Section
3417 12.2.6). The C1G2 specific InventoryCommand is already defined in Section 16.3.1.2.1.

3418

3419 **16.3.1.5 Reporting Parameters**

3420 This section of air protocol specific parameters corresponds to LLRP parameters
 3421 encodings specified in section 15.2.1.5.

3422 **16.3.1.5.1 C1G2EPCMemorySelector Parameter**

3423

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 348										Length															
C	P	Reserved																													

3424

3425 **Abbreviations**

3426

3427 C – EnableCRC

3428 P – EnablePCBits

3429

3430 See section 15.2.1.5.1.

3431

3432 **16.3.1.5.2 C1G2PC Parameter (TV-Encoding)**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=12						PC-Bits																								

3433

3434 See section 15.2.1.5.2.

3435 **16.3.1.5.3 C1G2CRC Parameter (TV-Encoding)**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=11						CRC																								

3436

3437 See section 15.2.1.5.3.

3438 **16.3.1.5.4 C1G2SingulationDetails Parameter (TV-Encoding)**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	Type=18						NumCollisionSlots										NumEmptySlots[15:8]														
NumEmptySlots[7:0]																															

3439

3440 See section 15.2.1.5.4.

3441 **16.3.1.5.5 C1G2 OpSpec Results**

3442 **16.3.1.5.5.1 C1G2ReadOpSpecResult Parameter**

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 349										Length															
Result						OpSpecID										ReadDataWordCount[15:8]															

3461

3462 **16.3.1.5.5.6 C1G2BlockWriteOpSpecResult Parameter**

0										1									2									3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 354										Length															
Result									OpSpecID												NumWordsWritten[15:8]										
NumWordsWritten[7:0]																															

3463

3464 See section 15.2.1.5.5.6.

3465 **16.4 Listing of Message and Parameter Types**

3466 This section lists the parameter and message types used in the binary encoding.

3467 **Table 4: Message Listing**

Message Name	Type
GET_READER_CAPABILITIES	1
GET_READER_CAPABILITIES_RESPONSE	11
ADD_ROSPEC	20
ADD_ROSPEC_RESPONSE	30
DELETE_ROSPEC	21
DELETE_ROSPEC_RESPONSE	31
START_ROSPEC	22
START_ROSPEC_RESPONSE	32
STOP_ROSPEC	23
STOP_ROSPEC_RESPONSE	33
ENABLE_ROSPEC	24
ENABLE_ROSPEC_RESPONSE	34
DISABLE_ROSPEC	25
DISABLE_ROSPEC_RESPONSE	35
GET_ROSPECS	26
GET_ROSPECS_RESPONSE	36
ADD_ACCESSSPEC	40
ADD_ACCESSSPEC_RESPONSE	50
DELETE_ACCESSSPEC	41
DELETE_ACCESSSPEC_RESPONSE	51
ENABLE_ACCESSSPEC	42
ENABLE_ACCESSSPEC_RESPONSE	52

DISABLE_ACCESSSPEC	43
DISABLE_ACCESSSPEC_RESPONSE	53
GET_ACCESSSPECS	44
GET_ACCESSSPECS_RESPONSE	54
CLIENT_REQUEST_OP	45
CLIENT_REQUEST_OP_RESPONSE	55
GET_REPORT	60
RO_ACCESS_REPORT	61
KEEPALIVE	62
KEEPALIVE_ACK	72
READER_EVENT_NOTIFICATION	63
ENABLE_EVENTS_AND_REPORTS	64
ERROR_MESSAGE	100
GET_READER_CONFIG	2
GET_READER_CONFIG_RESPONSE	12
SET_READER_CONFIG	3
SET_READER_CONFIG_RESPONSE	13
CLOSE_CONNECTION	14
CLOSE_CONNECTION_RESPONSE	4
CUSTOM_MESSAGE	1023

3468

3469

Table 5: Parameter Listing

Parameter Name	Type	TV-Encoded?
UTCTimeStamp	128	
Uptime	129	
GeneralDeviceCapabilities	137	
ReaderSensitivityTableEntry	139	
PerAntennaAirProtocol	140	
GPIOCapabilities	141	
LLRPCapabilities	142	
RegulatoryCapabilities	143	
UHFBandCapabilities	144	
TransmitPowerLevelTableEntry	145	
FrequencyInformation	146	
FrequencyHopTable	147	

FixedFrequencyTable	148	
PerAntennaReceiveSensitivityRange	149	
ROSpec	177	
ROBoundarySpec	178	
ROSpecStartTrigger	179	
PeriodicTriggerValue	180	
GPITriggerValue	181	
ROSpecStopTrigger	182	
AI Spec	183	
AI SpecStopTrigger	184	
TagObservationTrigger	185	
InventoryParameterSpec	186	
RFSurveySpec	187	
RFSurveySpecStopTrigger	188	
AccessSpec	207	
AccessSpecStopTrigger	208	
AccessCommand	209	
ClientRequestOpSpec	210	
ClientRequestResponse	211	
LLRPCConfigurationStateValue	217	
Identification	218	
GPOWriteData	219	
KeepaliveSpec	220	
AntennaProperties	221	
AntennaConfiguration	222	
RFReceiver	223	
RFTransmitter	224	
GPiPortCurrentState	225	
EventsAndReports	226	
ROReportSpec	237	
TagReportContentSelector	238	
AccessReportSpec	239	
TagReportData	240	
EPCData	241	
EPC-96	13	X

ROSpecID	9	X
SpecIndex	14	X
InventoryParameterSpecID	10	X
AntennaID	1	X
PeakRSSI	6	X
ChannelIndex	7	X
FirstSeenTimestampUTC	2	X
FirstSeenTimestampUptime	3	X
LastSeenTimestampUTC	4	X
LastSeenTimestampUptime	5	X
TagSeenCount	8	X
ClientRequestOpSpecResult	15	X
AccessSpecID	16	X
RFSurveyReportData	242	
FrequencyRSSILevelEntry	243	
ReadEventNotificationSpec	244	
EventNotificationState	245	
ReaderEventNotificationData	246	
HoppingEvent	247	
GPIEvent	248	
ROSpecEvent	249	
ReportBufferLevelWarningEvent	250	
ReportBufferOverflowErrorEvent	251	
ReaderExceptionEvent	252	
OpSpecID	17	X
RFSurveyEvent	253	
AISpecEvent	254	
AntennaEvent	255	
ConnectionAttemptEvent	256	
ConnectionCloseEvent	257	
LLRPStatus	287	
FieldError	288	
ParameterError	289	
Custom	1023	
C1G2LLRPCapabilities	327	

UHFC1G2RFModeTable	328	
UHFC1G2RFModeTableEntry	329	
C1G2InventoryCommand	330	
C1G2Filter	331	
C1G2TagInventoryMask	332	
C1G2TagInventoryStateAwareFilterAction	333	
C1G2TagInventoryStateUnawareFilterAction	334	
C1G2RFControl	335	
C1G2SingulationControl	336	
C1G2TagInventoryStateAwareSingulationAction	337	
C1G2TagSpec	338	
C1G2TargetTag	339	
C1G2Read	341	
C1G2Write	342	
C1G2Kill	343	
C1G2Lock	344	
C1G2LockPayload	345	
C1G2BlockErase	346	
C1G2BlockWrite	347	
C1G2EPCMemorySelector	348	
C1G2PC	12	X
C1G2CRC	11	X
C1G2SingulationDetails	18	X
C1G2ReadOpSpecResult	349	
C1G2WriteOpSpecResult	350	
C1G2KillOpSpecResult	351	
C1G2LockOpSpecResult	352	
C1G2BlockEraseOpSpecResult	353	
C1G2BlockWriteOpSpecResult	354	

3470

3471

3472 **17 Transmitter Behavior of a Reader**

3473 A Reader SHALL enable its transmitter only under the following conditions:

- 3474
 - When an ROSpec is in the active state.

3475 • Between a GET/SET_READER_CONFIG containing a RequestedData field with
3476 value 0 (All) or 2 (Antenna Properties) and the corresponding
3477 GET/SET_READER_CONFIG_RESPONSE.

3478 **18 Connection and Transport**

3479 The Reader SHALL maintain LLRP configuration state during an LLRP connection.

3480 The Reader MAY maintain configuration or data state when a connection fails, or across
3481 LLRP connections.

3482 **18.1 TCP Transport**

3483 LLRP end-to-end communications based on TCP/IP connections SHALL be
3484 implemented in accordance with the requirements specified in this section. These
3485 requirements are defined as the LLRP *TCP Transport*.

3486 Readers SHALL be able to both initiate and accept LLRP TCP connections. Readers
3487 MAY be configured such that, at any given time, they only either initiate or accept an
3488 LLRP connection. If so, the mechanism for configuring a Reader to either initiate or
3489 accept an LLRP connection is not specified by LLRP.

3490 Clients SHALL be able both to initiate and accept LLRP TCP connections. Clients MAY
3491 be configured such that, at any given time, they only either initiate or accept an LLRP
3492 connection. If so, the mechanism for configuring a Client to either initiate or accept an
3493 LLRP connection is not specified by LLRP.

3494 For Readers and Clients, that are configured to accept connections, the default port is
3495 5084, as established by IANA (see <http://www.iana.org/assignments/port-numbers>), but
3496 other ports can be used.

3497 When a TCP connection (called the *established connection*) is initiated by either the
3498 Reader or the Client, the Reader SHALL reply with a status report message before
3499 communicating any other information. This report's status parameter,
3500 ConnectionAttemptEvent, SHALL be set to indicate connection success (see section
3501 13.2.6.11). No other parameters may be contained within this message. The Client
3502 SHALL not send any information to the Reader until this status report message is
3503 received.

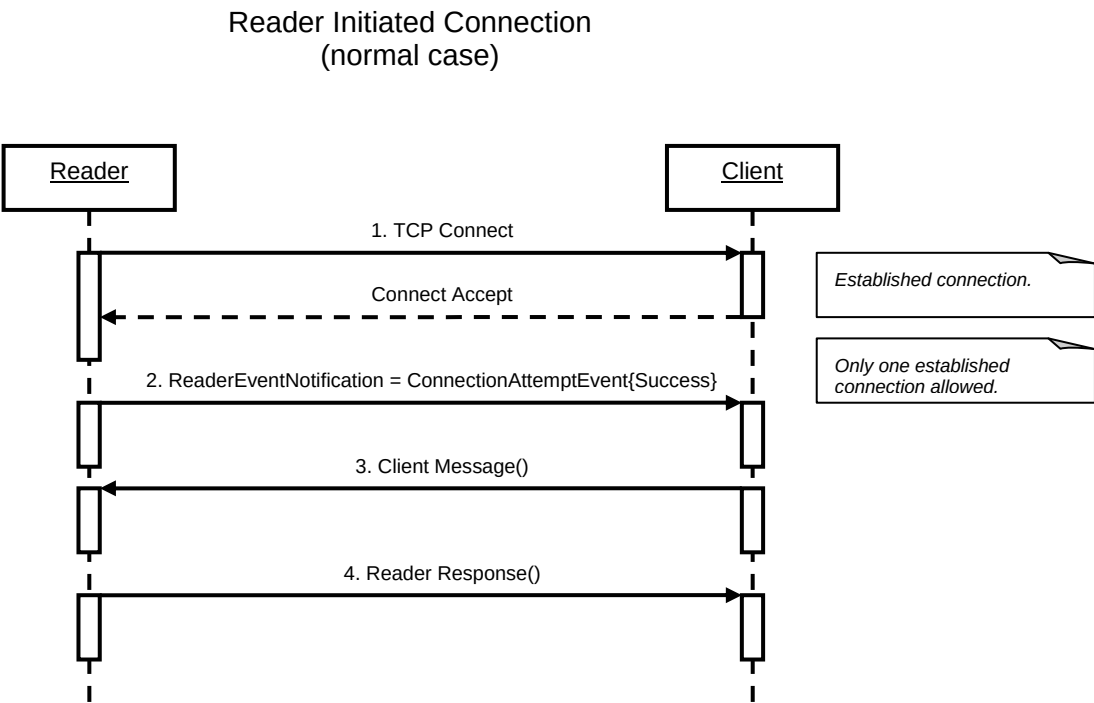
3504 Readers SHALL limit communications to a single established connection on a Reader IP
3505 address and TCP port. Readers MAY momentarily accept TCP connections (called
3506 *momentary connections*) in addition to the Reader's one established connection on a
3507 Reader IP address and TCP port. If a momentary connection is accepted, then the Reader
3508 SHALL send a status report message on the Reader's established connection. This
3509 report's status parameter, ConnectionAttemptEvent, SHALL be set to indicate that
3510 another connection was attempted (see section 13.2.6.11). If this action results in a TCP
3511 error, then the Reader MAY close the established connection and then treat the
3512 momentary connection as a new established connection. In this case, the Reader SHALL
3513 reply with a status report message on the newly created established connection, as
3514 specified above, indicating connection success.

3515 If the established connection is not closed, then the Reader SHALL reply on the
3516 momentary connection with a status report message. This report's status parameter,
3517 ConnectionAttemptEvent, SHALL be set to indicate connection failure. The Reader
3518 SHALL use the appropriate connection failed status value as defined in section 13.2.6.11.
3519 Once the connection failure message is sent, the Reader SHALL close the momentary
3520 connection.

3521 The following UML sequence diagrams illustrate different scenarios of a Reader and
3522 Client initiating TCP connections.

3523

3524



3525

3526 **Figure 14: Reader Initiated Connection (Normal)**

3527

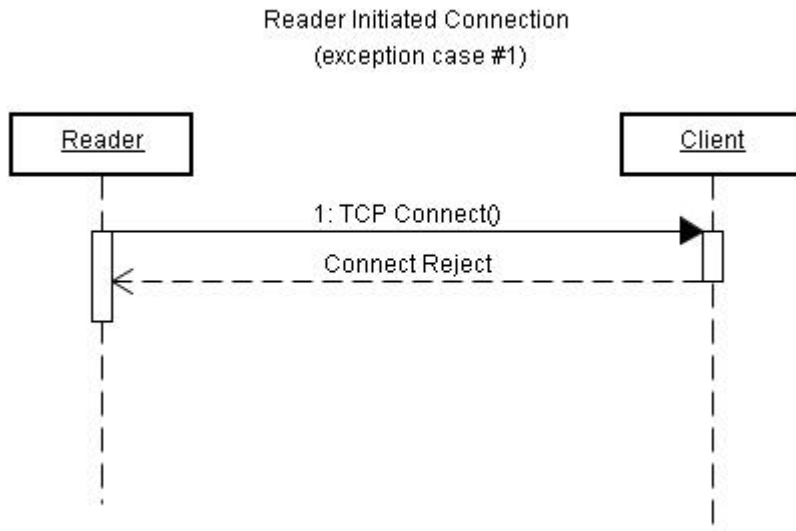


Figure 15: Reader Initiated Connection (Exception)

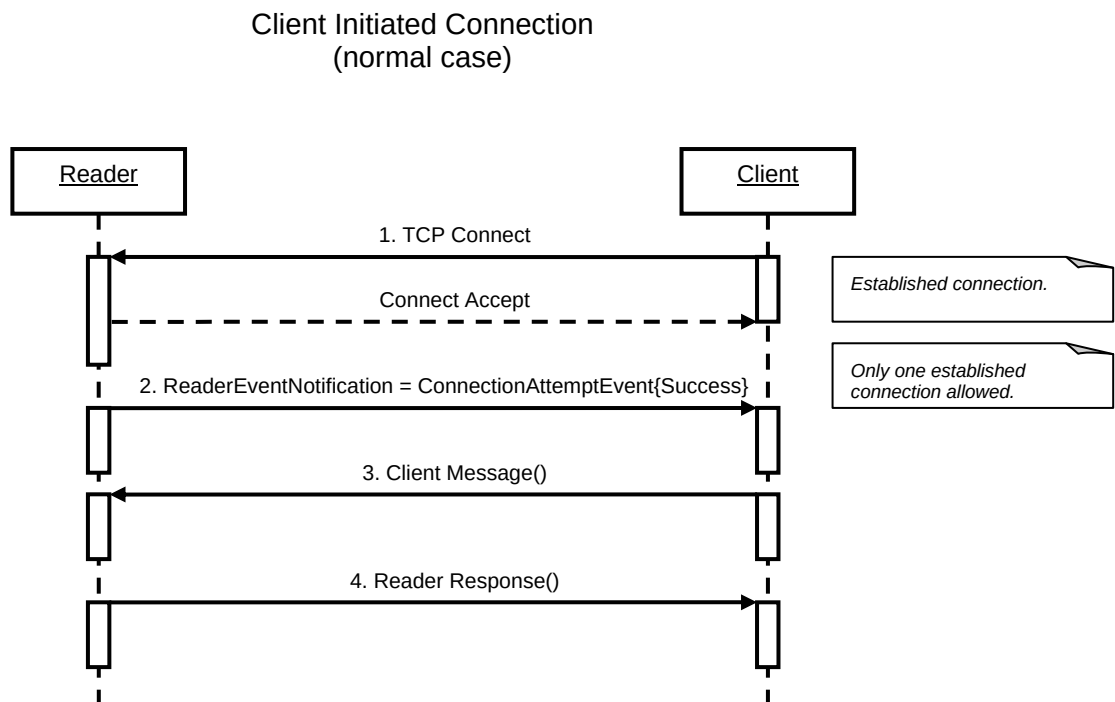


Figure 16: Client Initiated Connection (Normal)

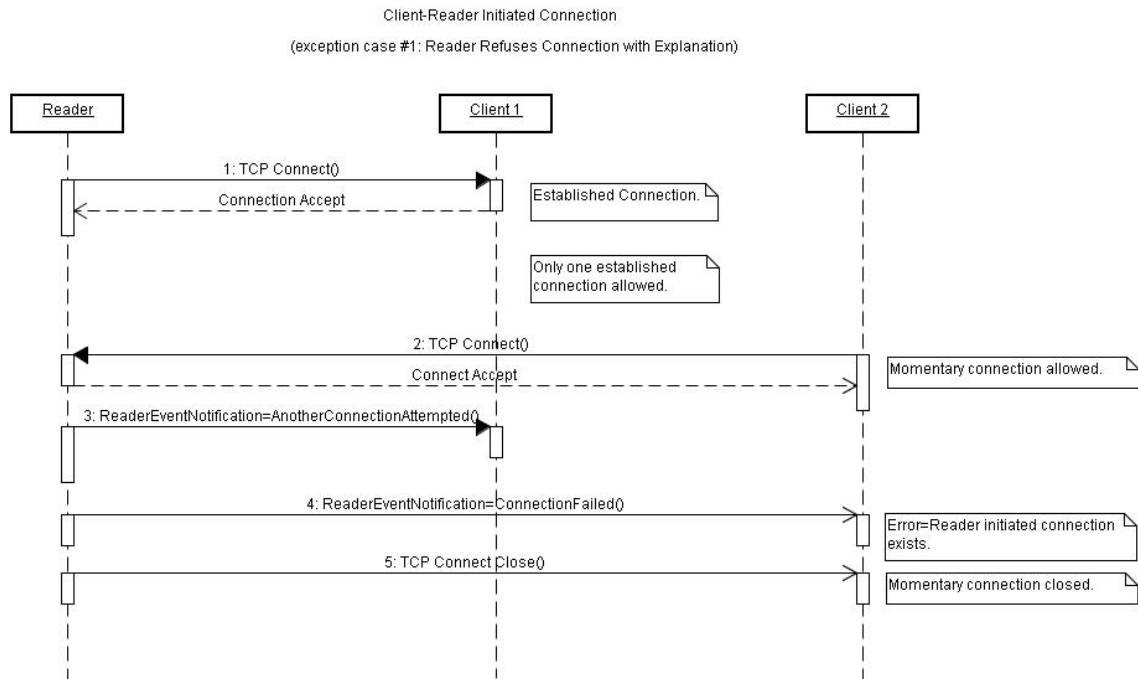


Figure 17: Client Initiated Connection (Exception #1)

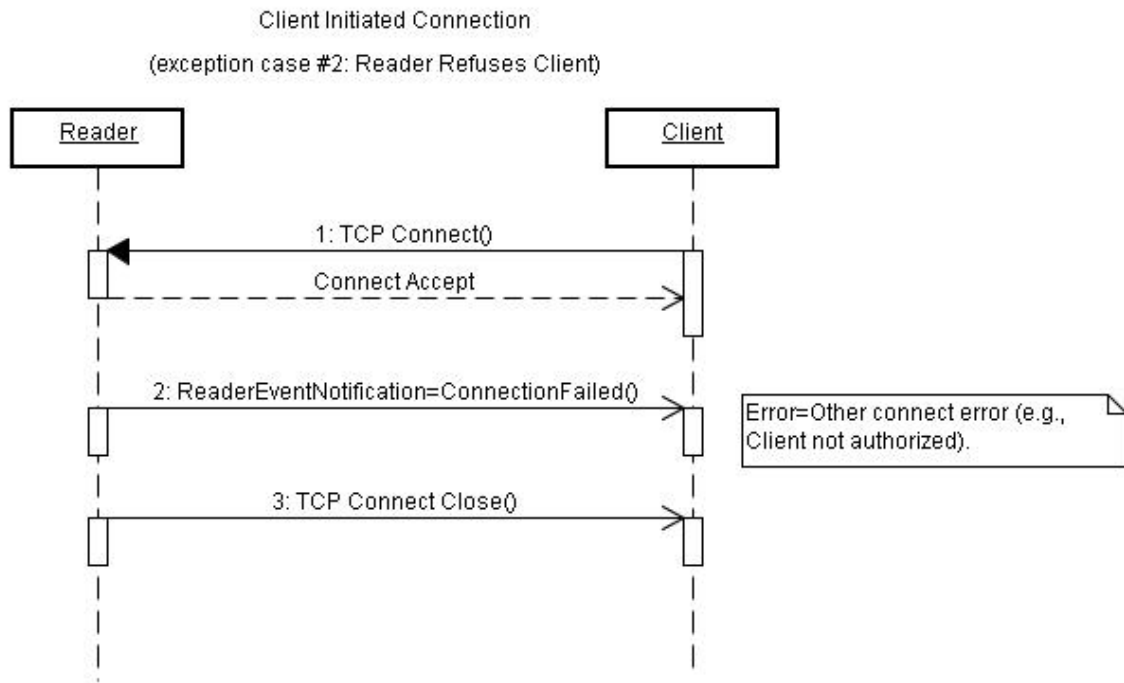


Figure 18: Client Initiated Connection (exception #2)

Client Initiated Connection
(exception case #3: Reader Refuses Another Connect)

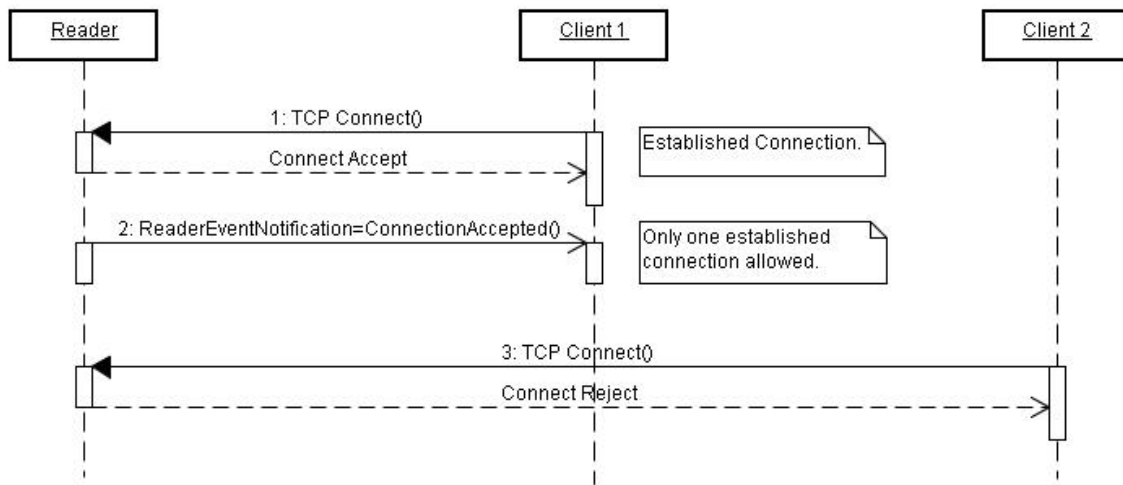


Figure 19: Client Initiated Connection (exception #3)

Client Initiated Connection
(exception case #4: New Established Connection)

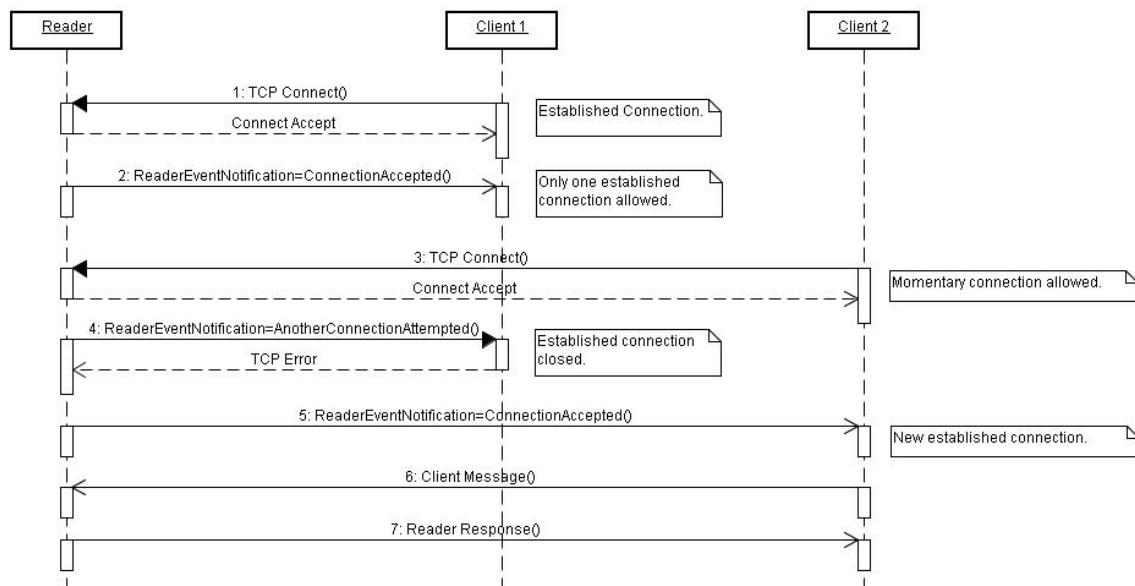


Figure 20: Client Initiated Connection (exception #4)

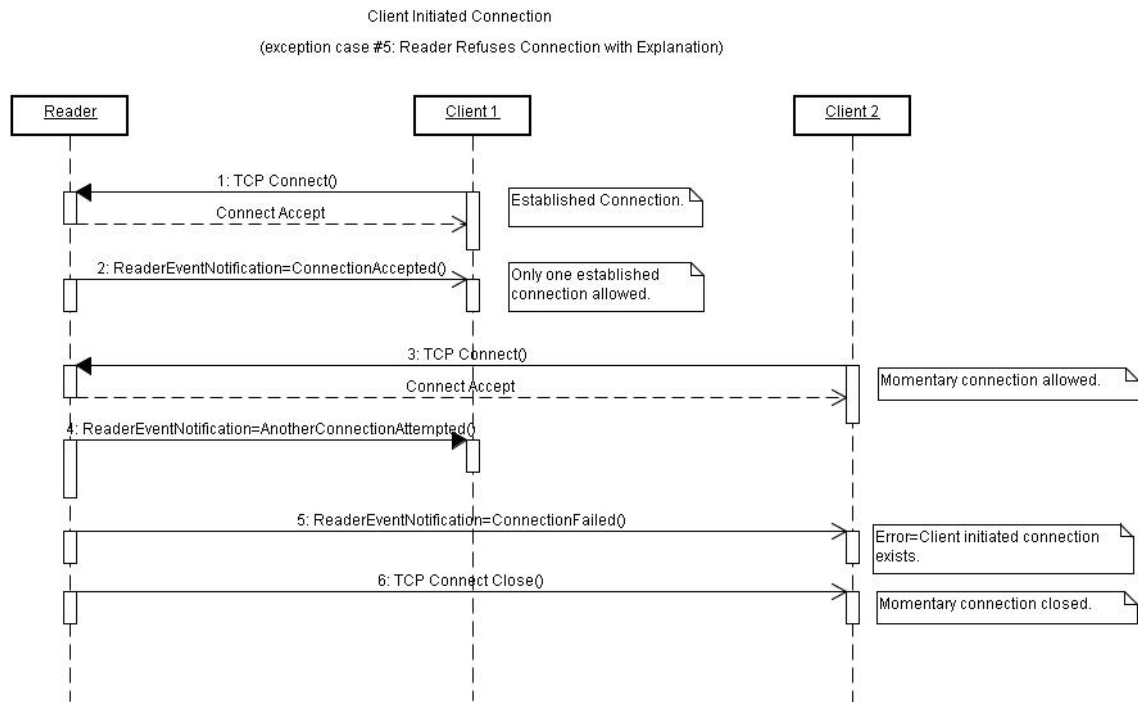


Figure 21: Client Initiated Connection (exception #5)

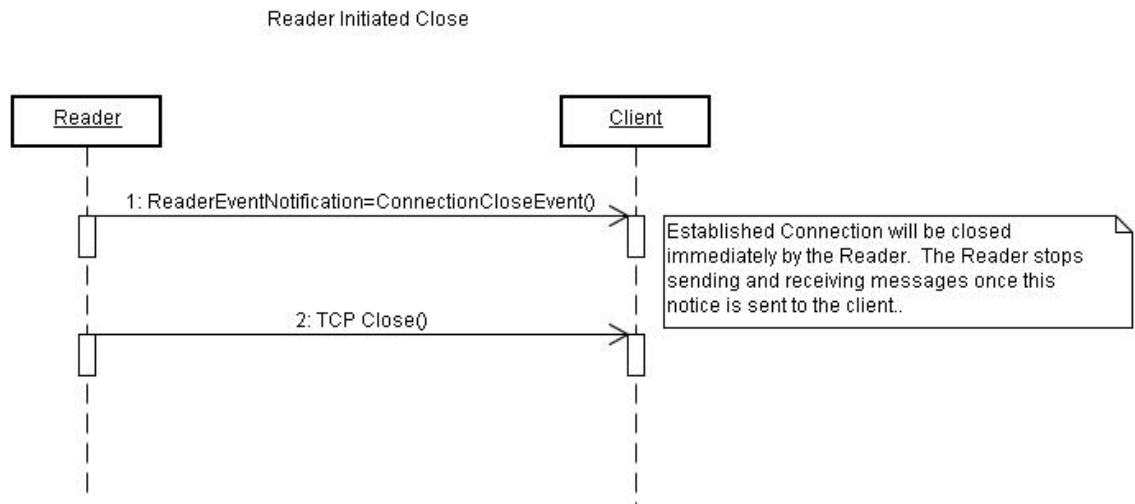


Figure 22: Reader Initiated Close

Client Initiated Close
(normal case)

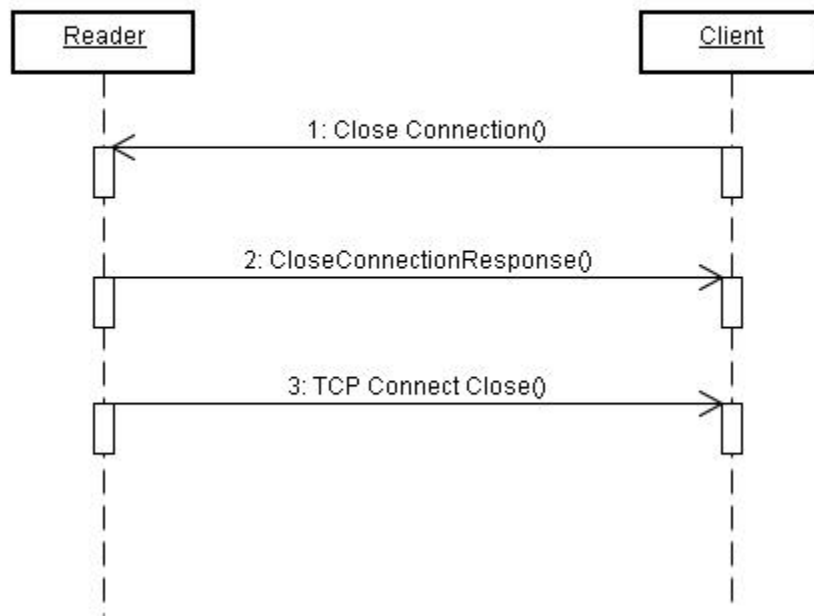


Figure 23: Client Initiated Close (Normal)

Client Initiated Close
(exception case #1: Reader Fails to Respond)

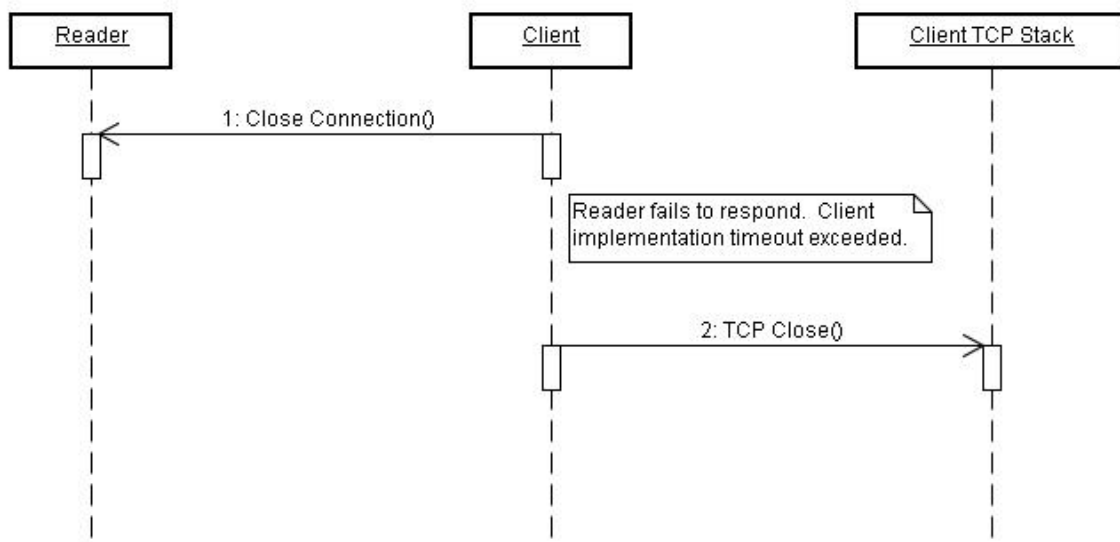


Figure 24: Client Initiated Close (Exception)

18.2 Security in TCP Transport

This section describes the security aspects for LLRP connections running over a TCP transport binding. Refer to the previous section for any TCP connection related requirements.

18.2.1 Normative Section

The LLRP Client and LLRP Reader MAY implement TLS. The LLRP Client and LLRP Reader MAY use a different port for TLS LLRP connections and non-TLS LLRP connections.

The LLRP Client MAY be capable of operating in a mixed deployment, where it communicates using TLS with a set of Readers and just plain TCP with a different set of Readers. In such mixed deployments, the LLRP Client MAY use different ports for TLS and non-TLS LLRP connections. The default port for TLS-LLRP connections is 5085, as established by IANA (see <http://www.iana.org/assignments/port-numbers>), but other ports can be used.

The LLRP endpoint that initiates the TLS connection MAY be the same LLRP endpoint that initiated the underlying TCP connection.

The LLRP endpoints SHALL use at least TLS1.0 [TLS10] and are recommended to use TLS1.1 [TLS11].

If the Reader or Client uses X.509 certificates[X509] for authentication, the certificates SHALL be compliant with the EPCGlobal Security2 working group specification [SEC2].

18.2.2 Informative Section

18.2.2.1 Overview of TLS

The TLS protocol provides privacy and data integrity between two authenticated communicating applications. TLS is a light weight transport protocol and has been proven to be reliable and secure by the use of millions of real users for many years. The strength of TLS can be chosen by the cipher suite negotiated by the two communicating parties through a flexible mechanism during the handshaking.

TLS is particularly useful for TCP based applications. First, a TLS client initiates a connection with the TLS server. After a TLS connection is established, the applications can use the transport connection like an ordinary TCP connection, while having the added value that the data is protected and that both parties are mutually authenticated.

For interoperability, a TLS client and server have to implement at least one common cipher suite. The credentials required for mutual authentication depend on the suite negotiated. For example, if the negotiated suite is using RSA for key exchange, then the server must own a server certificate (with private key) for RSA encryption purposes while the client must have a client certificate (with private key) for RSA signing purposes. Further, each side must have the root Certificate Authority (CA) certificates to

3599 verify the certificates presented by the peers. TLS also requires each party to present the
3600 CA certificates (except the root) that directly and indirectly issue the certificate.

3601 **18.2.2.2 Threat Analysis for LLRP**

3602 With TLS being used for Reader and Client communication, the following protections are
3603 provided, assuming that the credentials for the TLS client and server are not stolen:

- 3604 • Readers only talk to authorized LLRP Clients;
- 3605 • LLRP Clients only talk to authorized Readers;
- 3606 • No other party can read the LLRP messages (privacy protection) or inject/modify
3607 messages without being detected (integrity protection).

3608 Note that the strength of protection depends on the negotiated cipher suite.

3609 **18.2.2.3 Configuration Elements for TLS**

3610 In order to use TLS for LLRP, the following information has to be configured and/or
3611 provisioned at each entity (Reader or Client):

- 3612 • **TLS enabled:** Yes or no. If TLS is not enabled, the rest of the information need
3613 not be configured and the LLRP endpoint (Reader or Client) SHALL use TCP
3614 directly.
- 3615 • **TLS role:** Whether the LLRP endpoint is playing the TLS client or the TLS
3616 server role. A TLS client initiates a TCP connection to jump start TLS
3617 handshaking. A TLS server passively listens on the TCP server port.
- 3618 • **Preferred list of cipher suites:** A TLS client proposes the list of cipher suites to
3619 the TLS server during TLS handshaking. The TLS server will pick one suite from
3620 the proposed list if it is also in the preferred list maintained by the server. In TLS,
3621 the order of suites in the proposed list has no significance. Also, it is up to the
3622 server's local policy to select when there are multiple choices.
- 3623 • **Certificates and private keys:** A TLS server needs a server certificate (with
3624 private key) for TLS server authentication. A TLS client needs a client certificate
3625 (with private key) for TLS client authentication. In each case, all the CA
3626 certificates (except the root) in the chain have to be available.
- 3627 • **Root CA certificates:** A TLS server needs to maintain the root CA certificate of
3628 the client certificate. This is used for verifying client certificates. A TLS client
3629 needs to maintain the root CA certificate of the server certificate. This is used for
3630 verifying server certificates.
- 3631 • **List of authorized devices:** Each TLS server MAY have a list of authorized TLS
3632 clients that can connect to it. Likewise, each TLS client MAY have a list of
3633 authorized TLS servers that it can connect to.

3635 The configuration and/or provisioning of a LLRP endpoint is out of the scope of TLS and
3636 LLRP. Provisioning is important but does not affect the interoperability of LLRP.
3637 Vendors should have the flexibility to choose the most cost-effective ways (for
3638 provisioning and protecting provisioned credentials) based on designs, available
3639 technologies, potential threats, security requirements, and so on. This is a topic that
3640 should be addressed in DCI.

18.2.2.4 Why different TLS server port?

It is recommended that the TLS server should listen to a TCP port different from that for non-TLS mode for the following reasons:

- If one of the endpoints has to be deployed behind firewalls, IT managers are more willing to open a port they know only TLS traffic can pass through.
- Without using a different port, a non-TLS server may be confused by the TLS Client-Hello handshaking message.
- Without using a different port, a TLS server may be confused by the LLRP application message (non-TLS handshaking message).
- Without using a different port, for each new TCP connection, a server in a mixed environment (TLS and non-TLS) may have to wait a few moments to see if a Client-Hello message ever arrives from the client before it can conclude whether it is a TLS connection or not.
- Without using a different port, it is potentially harder to implement a hybrid server if the server relies on third-party libraries for handling TLS. This is because the server application has to read the first message from the client to know if it is a TLS connection. It may be difficult for the TLS library to take over a connection after the TLS Client-Hello message has been consumed.

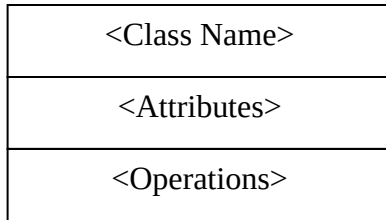
However, if a deployment in totality is only TLS or only non-TLS, the LLRP endpoint can be configured only as a TLS server or non-TLS server exclusively, then there should be no problem using the same port, as long as a non-TLS server can ignore TLS handshaking messages from a TLS client and as long as a TLS server can ignore non-TLS handshaking messages from a non-TLS client.

19 (Informative) Object Model

The Object Model (OM) presented in this section illustrates the data structures inherent in the LLRP specification and further described in section 5. These OM diagrams are based upon Unified Modeling Language (UML) notation (see www.uml.org). There are two kinds of LLRP data structures: 1) *messages* and 2) *data parameters*. Messages can be composed of data parameters. Data parameters can be further composed of other data parameters. A simple data element (i.e., a data element with no subcomponents) is called a *data field*.

In the OM, both kinds of data structures are represented by UML class diagrams. Data fields are represented as class attributes.

A UML class is defined as a collection of objects with common structure, common relationships, etc. A UML class is illustrated as a rectangle partitioned into three compartments as follows:



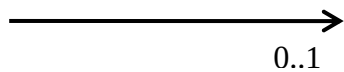
3678

3679 The OM is used only to describe structure and therefore the *Operations* compartment is
3680 left empty for all OM classes.

3681 There are three class relationship notations used in the OM:

3682 - Association with one-way navigation.

3683



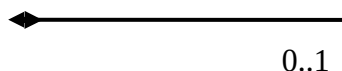
3684

3685 This notation represents that one class includes a reference to another class (the
3686 arrowhead side). LLRP data structures reference each other via an identifier (e.g.,
3687 ROSpecID).

3688 A number or a range of numbers (e.g., 0..1) can appear on either side of the line.
3689 This is the multiplicity of the relationship (e.g., the number of instances of one
3690 class related to one instance of the other class). If no number appears on a side of
3691 the line, then a one is implied.

3692 - Aggregation.

3693



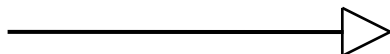
3694

3695 This notation represents that one class (the diamond side) includes another class
3696 embedded within it.

3697 A number or a range of numbers (e.g., 0..1) can appear on either side of the line.
3698 This is the multiplicity of the relationship (e.g., the number of instances of one
3699 class related to one instance of the other class). If no number appears on a side of
3700 the line, then a one is implied.

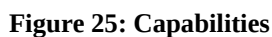
3701 - Inheritance.

3702

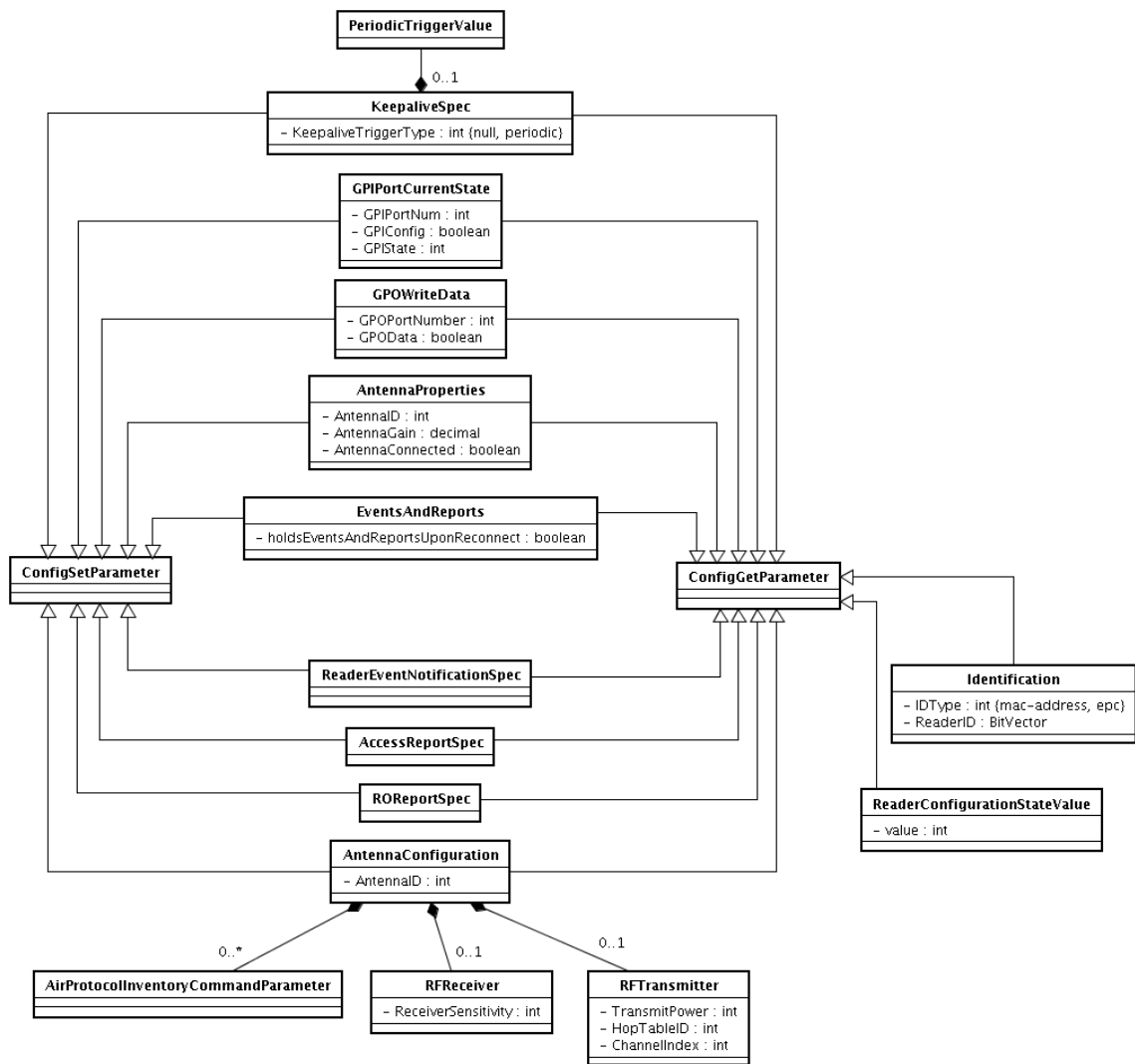


3703 This notation represents that one class is a superclass of another class known as
3704 the base class (the arrowhead side). A superclass includes all attributes and
3705 relationships of the base class plus additional features.

3707
3708



3709 **19.2 Configuration**



3710
3711 **Figure 26: Configuration**

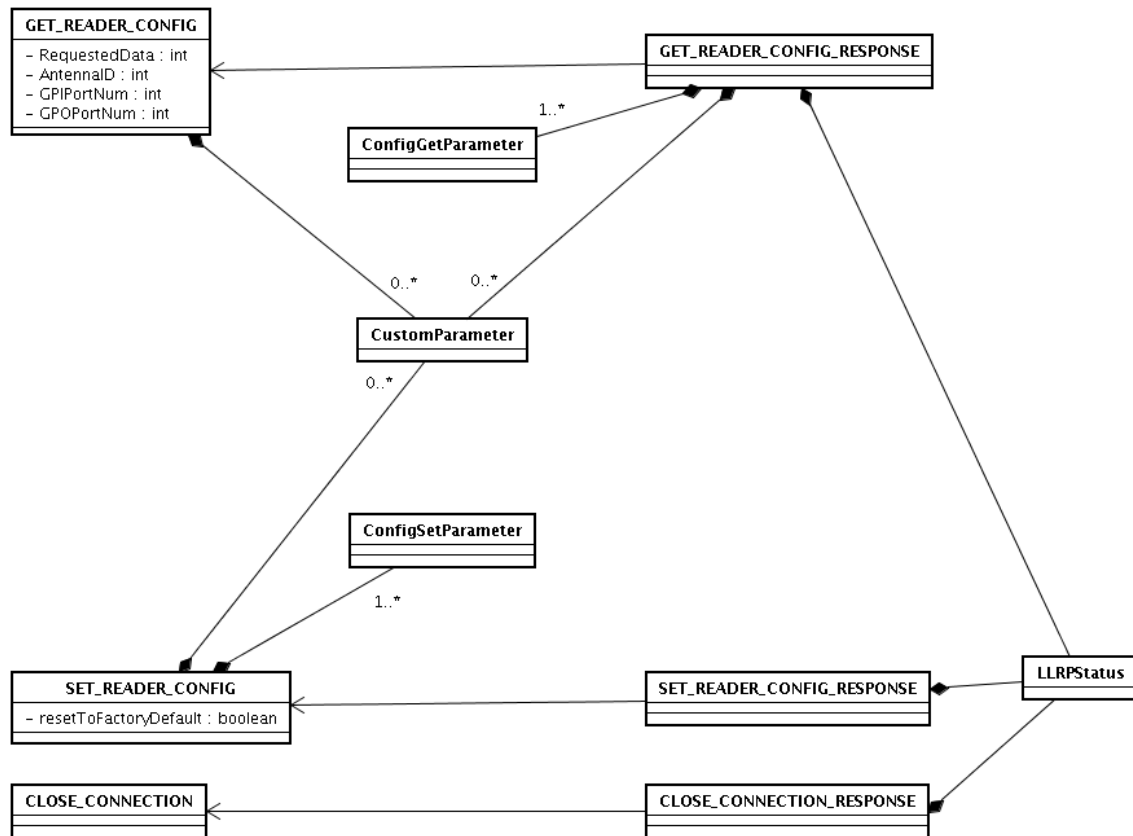


Figure 27: Configuration Commands

3714

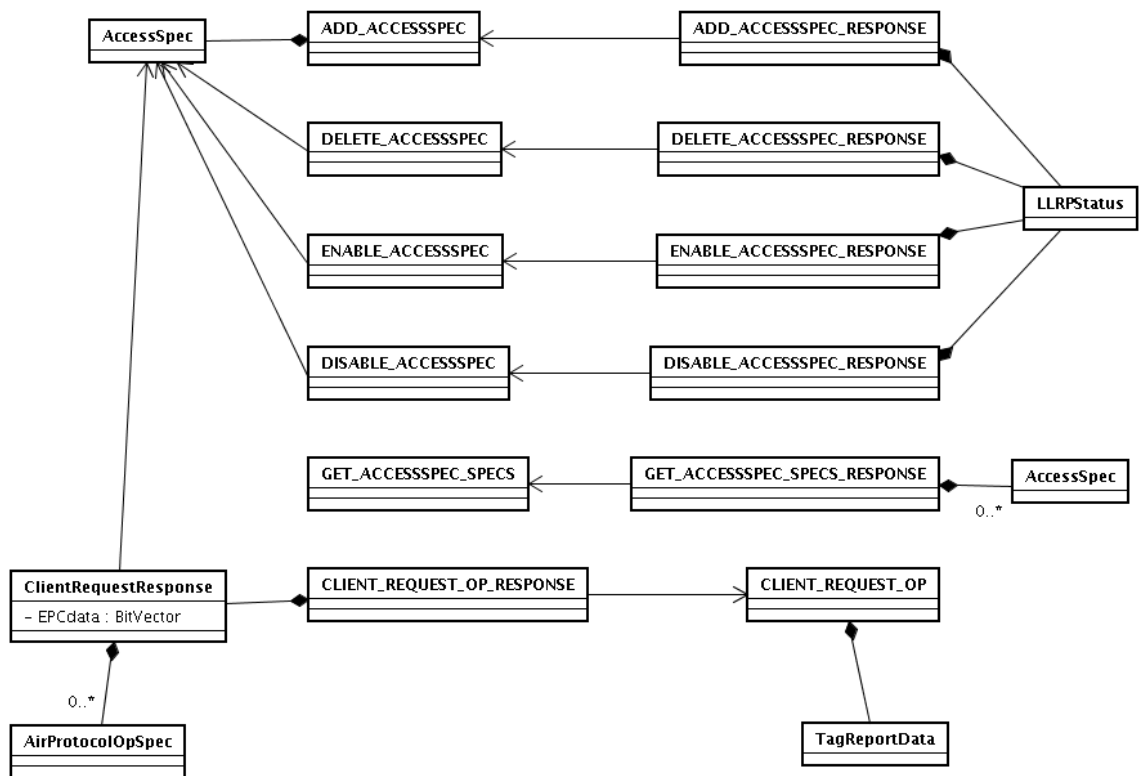


3716

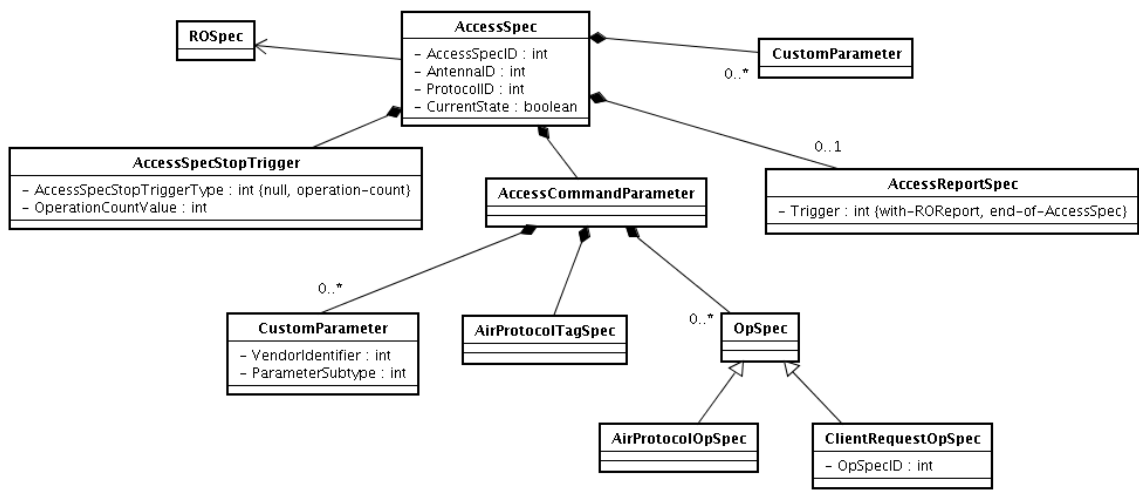


3718

3719 **19.4 AccessSpec**

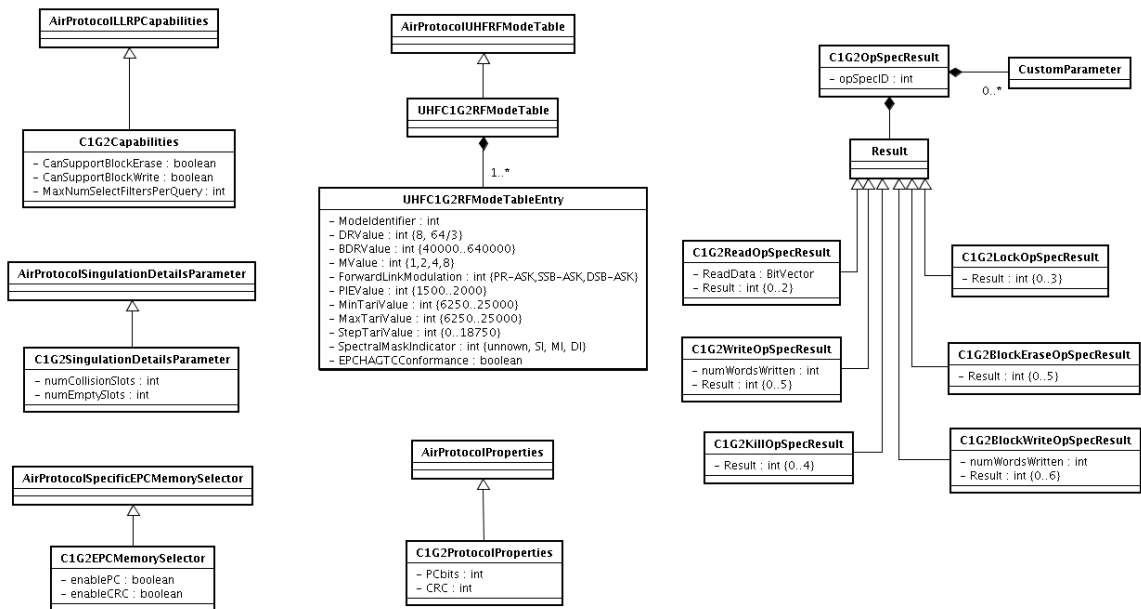


3720
3721 **Figure 30: Access Commands**

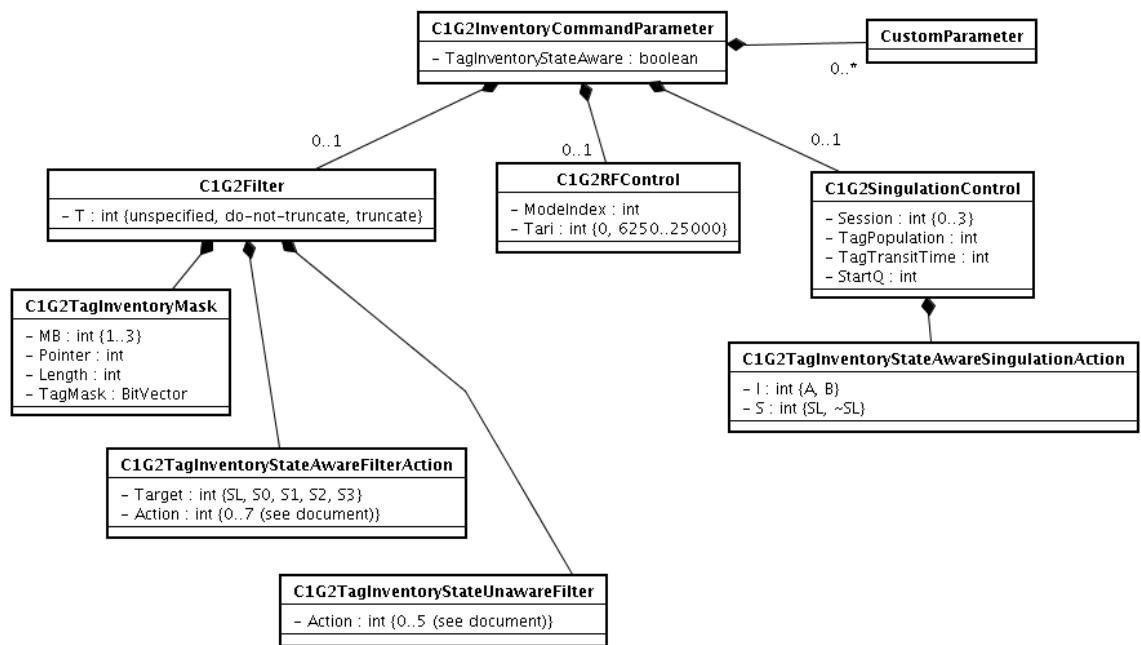


3722
3723 **Figure 31: AccessSpec**

3724 **19.5 C1G2 Parameters**



3725
3726 **Figure 32: C1G2 Parameters**



3727
3728 **Figure 33: C1G2 Inventory Command**

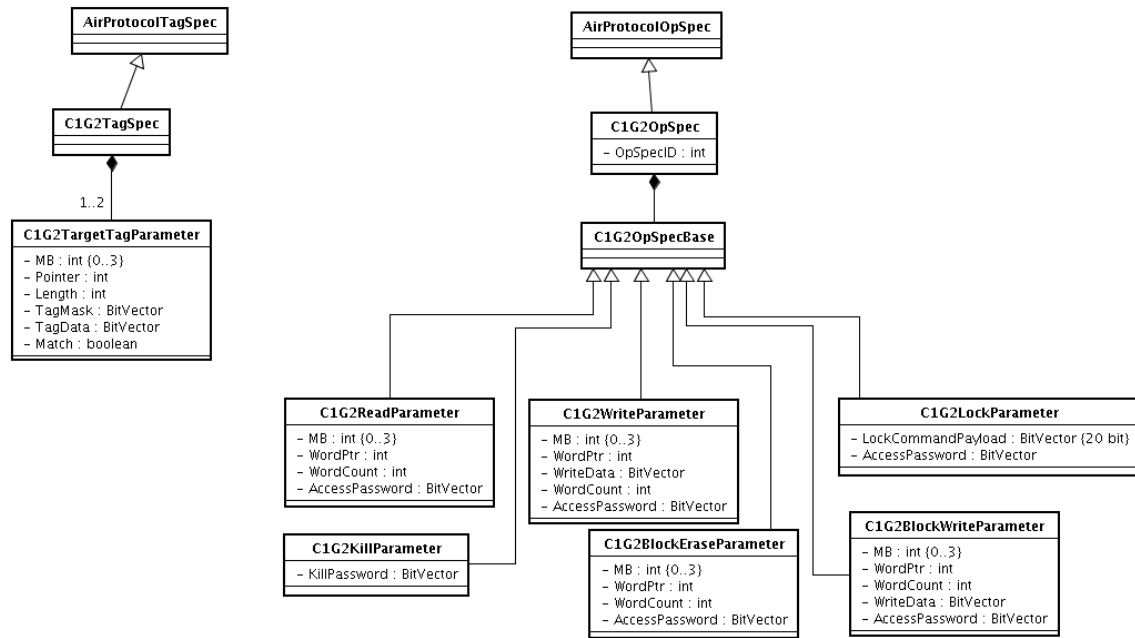


Figure 34: C1G2 AccessSpec

19.6 Reporting and Notification

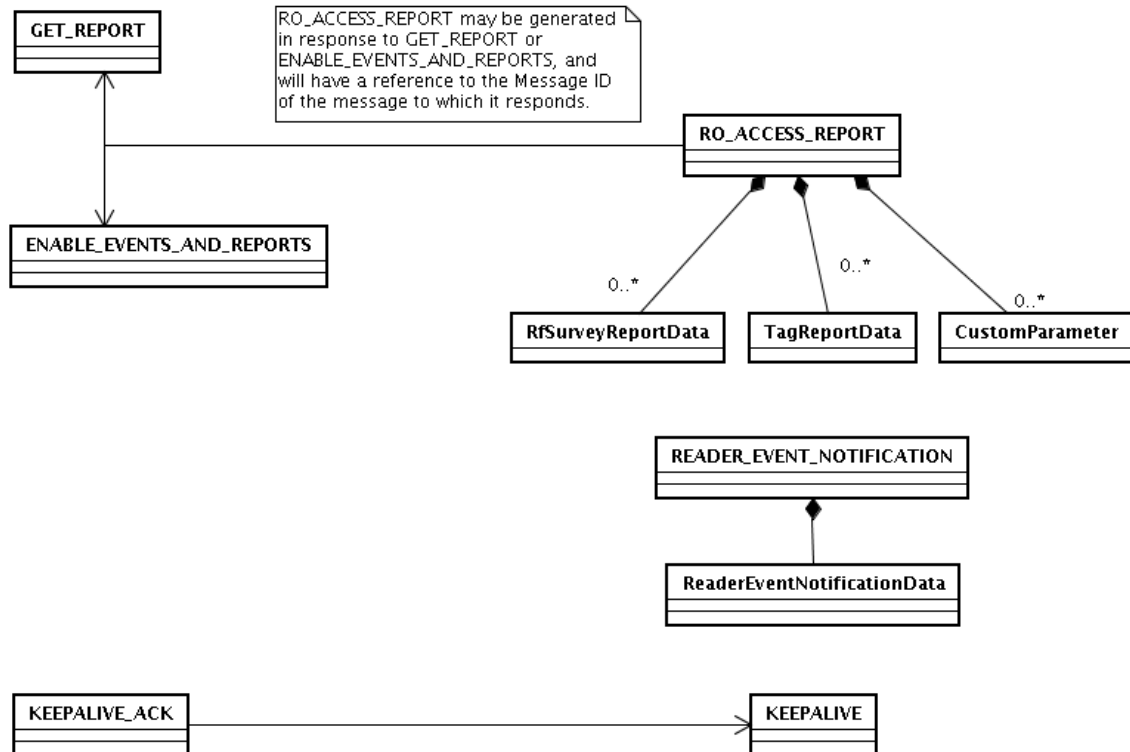


Figure 35: Reporting and Notification

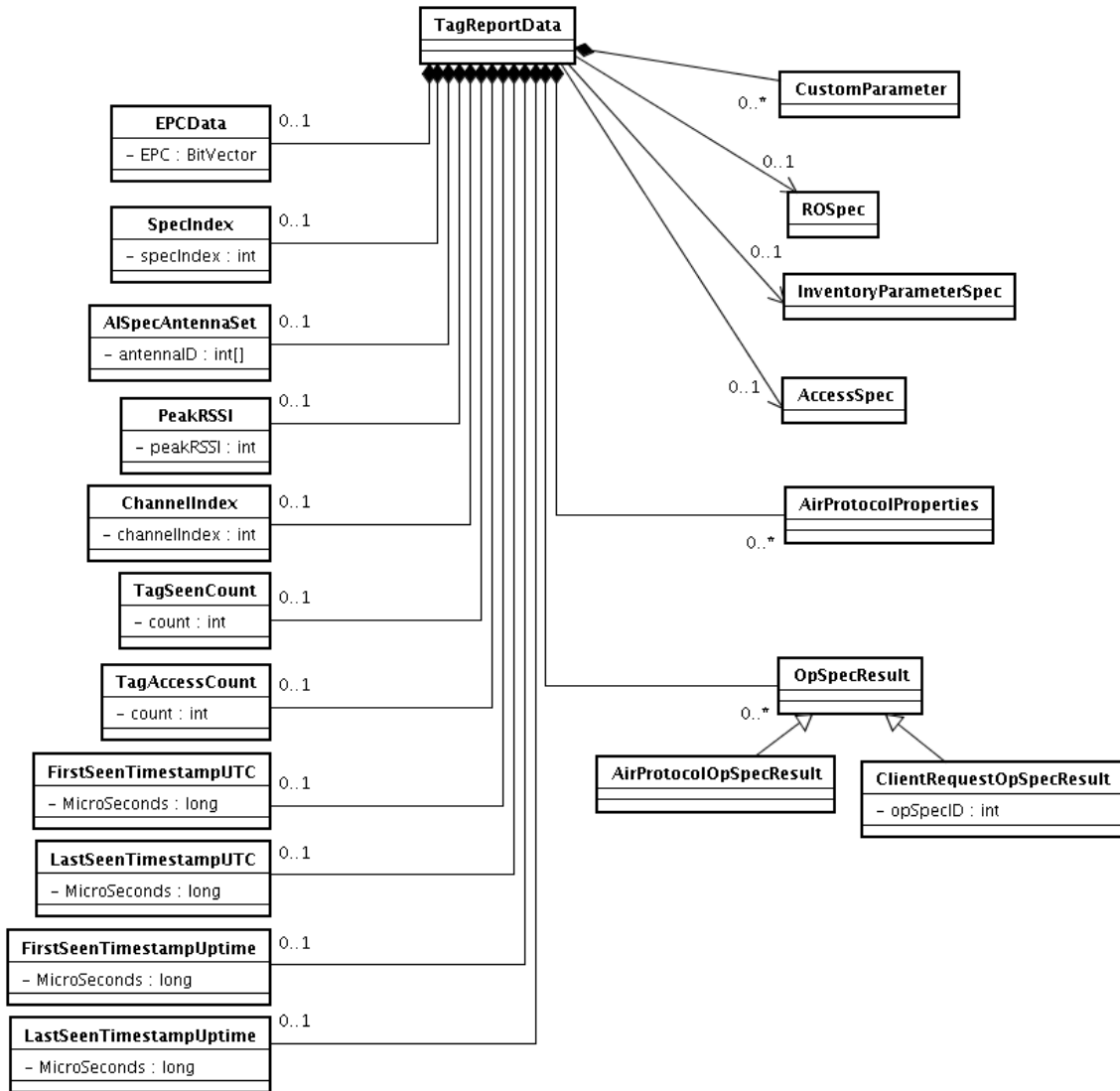


Figure 36: TagReportData

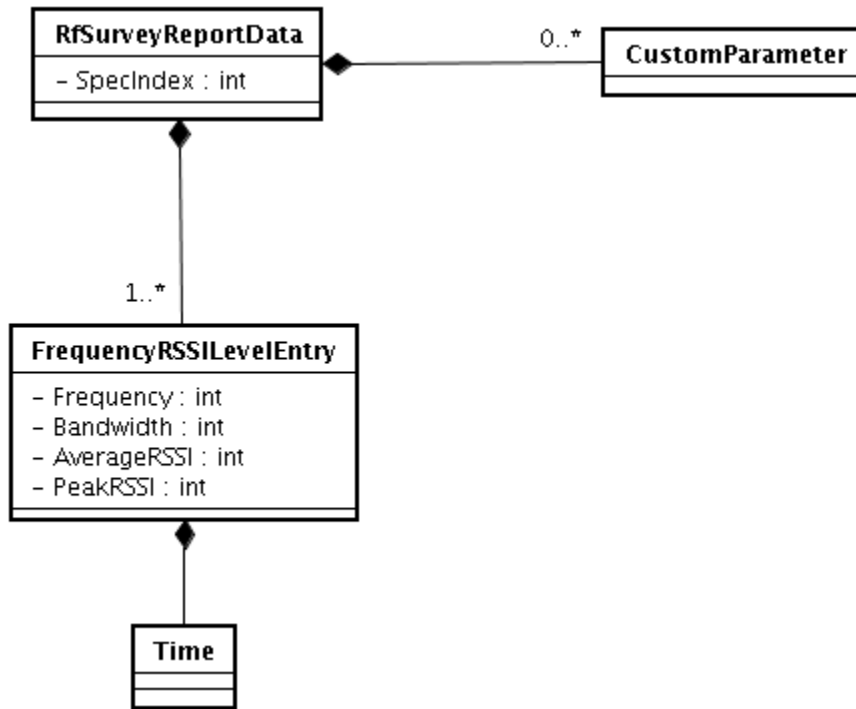


Figure 37: RfSurveyReportData

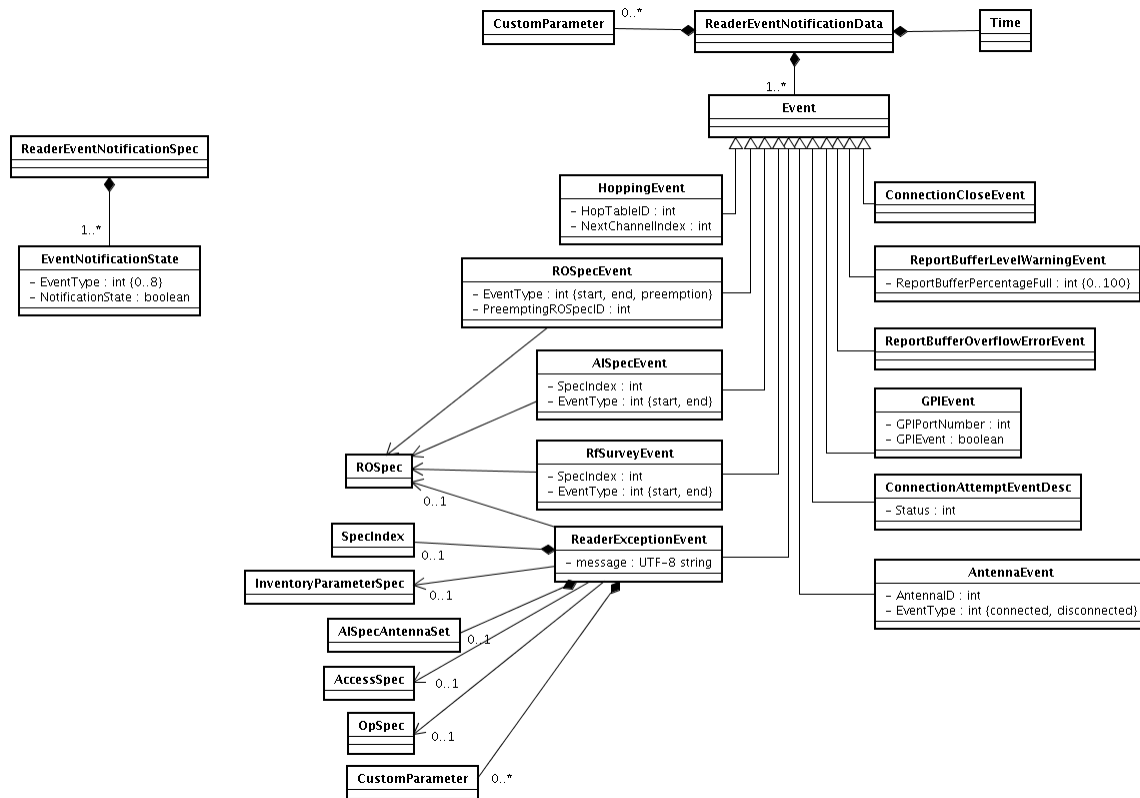
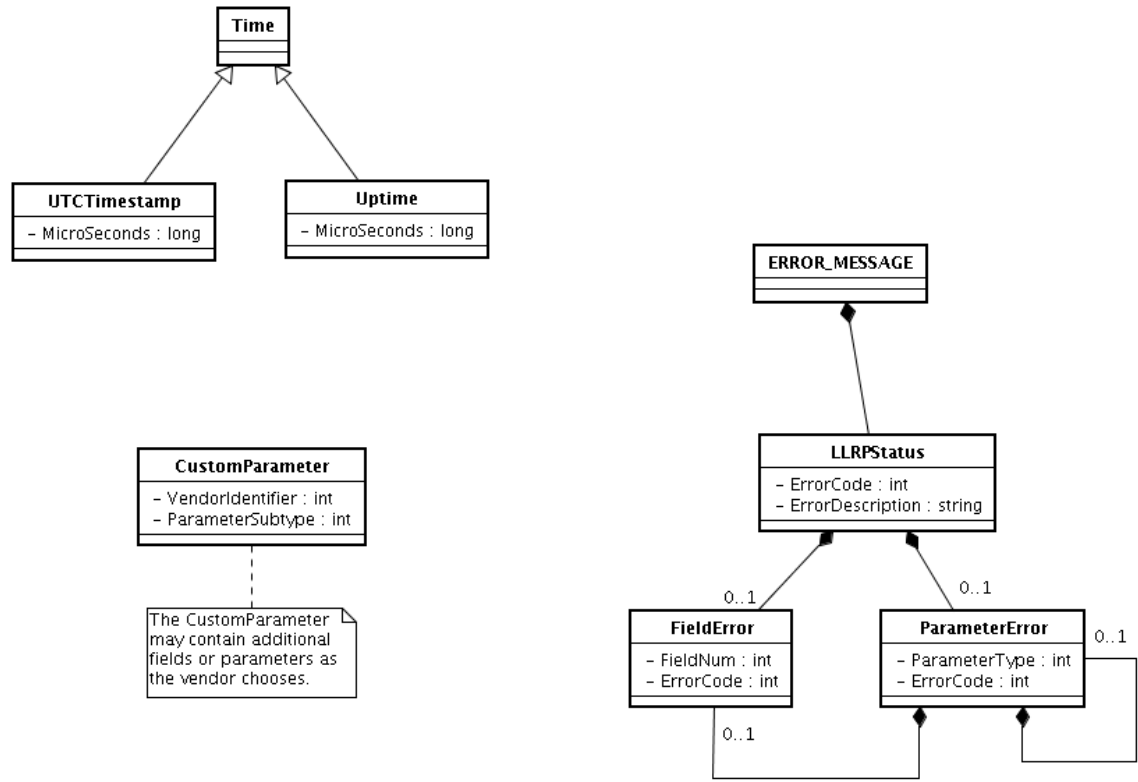


Figure 38: Reader Event Notification Data

3742 **19.7 General**



3743
3744 **Figure 39: General Data**

3745 **20 (Informative) TCP Keepalives**

3746 The TCP specification doesn't specify any specific handling of idle connections, where
3747 there is no data being transmitted by either end for a prolonged period of time. However,
3748 in some TCP implementations, there is an option called TCP-keepalive which may be
3749 turned on. If turned on, TCP-keepalive packets are sent only during periods of inactivity,
3750 on a configurable interval. If the connection is still valid, the other end responds with a
3751 segment containing an ack. If the connection is not valid the other end will reply with a
3752 connection reset (RST) and the connection is closed by this end.

3753 Due to events like network failures, or Client failures, half connections may remain at the
3754 Reader because the TCP connection was not cleanly terminated. If the Reader doesn't
3755 implement TCP-keepalive, the only way to recover (i.e., reconnect to the Reader) may be
3756 to reboot the Reader.

3757 However, there are Readers for which intermittent connectivity may be a normal mode of
3758 operation – e.g., mobile Readers, handheld Readers. When connectivity is lost for these
3759 devices, the use of TCP-keepalive acts negatively and closes the TCP session
3760 prematurely before the TCP session would have timed out. If keepalives were not used,
3761 the mobile Reader would just start sending LLRP messages as soon as the link layer is re-
3762 established without requiring a re-establishment of the TCP session as long as the TCP
3763 session did not timeout.

21 (Informative) References

[ISODir2] ISO, “Rules for the structure and drafting of International Standards (ISO/IEC Directives, Part 2, 2001, 4th edition),” July 2002.

[C1G2] Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9

http://www.epcglobalinc.org/standards_technology/EPCglobal2UHF RFIDProtocolV109122005.pdf

[ALE 1.0] EPCglobal Application Level Events (ALE) 1.0 Specification

http://www.epcglobalinc.org/standards_technology/EPCglobal_ApplicationALE_Specification_v112-2005.pdf

[Class0] 900 MHz Class 0 Radio Frequency (RF) Identification Tag Specification,

http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class0.pdf

[Class1] 860MHz -- 930 MHz Class 1 Radio Frequency (RF) Identification Tag Radio Frequency & Logical Communication Interface Specification.,

http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class1.pdf

[ARC] EPCglobal Architecture Framework 1.0

http://www.epcglobalinc.org/standards_technology/Final-epcglobal-arch-20050701.pdf

[ISO3166] ISO, “Codes for the Representation of Names of Countries”.

<http://www.iso.org/iso/en/prods-services/iso3166ma/index.html>

[UTC] IETF RFC 3339, “Date and Time on the Internet: Timestamps”, July 2002.

<http://www.ietf.org/rfc/rfc3339.txt>

[EUI64] "Guidelines For 64-bit Global Identifier (EUI-64)",

<http://standards.ieee.org/db/oui/tutorials/EUI64.html>

[TLS11] IETF RFC 4346, “The Transport Layer Security (TLS) Protocol Version 1.1”.

<http://www.ietf.org/rfc/rfc4346.txt>

[TLS10] IETF RFC 2246, “The TLS Protocol Version 1.0”,

<http://www.ietf.org/rfc/rfc2246.txt>

[X509] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - "The Directory Authentication Framework". 1988.

[SEC2] EPCGlobal Certificate Profile,

http://www.epcglobalinc.org/standards_technology/SAG%20Security%20Cert%20Profile%20March%208%202006.pdf

[UTF8] IETF RFC 3629, “UTF-8, a transformation format of ISO 10646”, November 2003. See also Annex D of ISO 10646-1:2000. <http://www.ietf.org/rfc/rfc3629.txt>

[HK] OFTA 1049,

<http://www.ofta.gov.hk/en/standards/hktaspec/hkta1049.pdf#search=%22OFTA%201049%22>

3800 [TW] DGT LP0002,
3801 <http://www.dgt.gov.tw/English/Type-approval/8.4/LP0002/LP0002-940324.pdf>
3802 [JPN] ARIB STD T89, http://www.arib.or.jp/english/html/overview/st_j.html
3803 [KOR] MIC Article 5-2, (Radio Equipment for RFID/USN): Technical Requirements
3804 for the radio equipment for passive RFID using the frequency range of 908.5~914MHz.
3805

22 Acknowledgement of Contributors and Companies Opt'd-in during the Creation of this Standard (Informative)

Disclaimer

Whilst every effort has been made to ensure that this document and the information contained herein are correct, EPCglobal and any other party involved in the creation of the document hereby state that the document is provided on an "as is" basis without warranty, either expressed or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights, of accuracy or fitness for purpose, and hereby disclaim any liability, direct or indirect, for damages or loss relating to the use of the document.

Below is a list of more active participants and contributors in the development of LLRP 1.0. This list does not acknowledge those who only monitored the process or those who chose not to have their name listed here. Active participants status was granted to those who generated emails, attended face-to-face meetings and conference calls that were associated with the development of this Standard.

First Name	Last Name	Company	Notable Role
Dave	Husak	Reva Systems	Co-Chair
Rob	Buck	Intermec	Co-Chair
Pattabhiraman	Krishna	Reva Systems	Editor
Mark	Frey	EPCglobal Inc.	Facilitator for WG
Software Team at Impinj		Impinj	Minutes Recorder
Marc	Horowitz	BEA Systems	
Suresh	Bhaskaran	Intellex	
Daniel	Paley	Tagent Corp.	
Bud	Biswas	Polaris Networks	
Bob	O'Hara	Cisco	
Daniel	Bowman	Kimberly-Clark Corp	
Margaret	Wasserman	ThingMagic, LLC	
Arthur	Howarth	Cisco	
Richard	Bach	GlobeRanger	
Rick	Schuessler	Symbol Tech./Motorola	

Howard	Kapustein	Manhattan Associates	
David	Missimer	Sirit	
Darrel	Pinson	Symbol Technologies, Inc.	
Matt	Poduska	Intermec	
Steve	Lockhart	Sirit	
David	Lavin	IBM	
Lynn	Hingst	Intermec	
John	Walter	Intermec	
Soumya	Roy chowdhury	Polaris Networks	
Martin	Jackson	Wal-Mart	
Steve	Lin	Sirit	
Bryan	Tracey	GlobeRanger	
Scott	de Deug	IBM	
Ted	Osinski	MET Labs	
Scott	Barvick	Reva Systems	
Manpreet	Singh	Symbol Technologies, Inc.	
Heena	Nandu	Intelleflex	
Gerhard	Gangl	7iD (formerly EOSS GmbH)	
Bill	Bares	Intelleflex	
Jim	Sykes	Savi Networks	
Sudhir	Hasbe	Sirit	
Albert	Lin	WJ Co.	
Shigeya	Suzuki	Auto-ID Labs - Japan	
Gay	Whitney	EPCglobal Inc.	
Jim	Reed	MET Labs	
Matthew	Harmon	Q.E.D. Systems	
Ricardo	Labiaga	Sun Microsystems	
Mark	Richardson	ThingMagic, LLC	
David	Nesbitt	Vue Technology	
Roger	Stewart	Applied Wireless (AWID)	
Yukiko	Yumoto	Auto Id Lab Japan	
Abel	Sanchez	Auto-ID Labs - MIT	
John	Williams	Auto-ID Labs - MIT	

Mark	Sompel	AWID	
Ken	Traub	BEA Systems	
Matt	Robshaw	France Telecom	
Wayne	Liu	Impinj	
Tareef	Nandu	Intelleflex	
Joe	Kubler	Intermec	
John	Walter	Intermec	
Peter	Anderla	KCC	
John	Boulas	KCC	
John	Anderla	KCC	
Moon Suk	Kim	Metarights	
Chang Yeol	Lee	Metarights	
Jens	Kungl	Metro	
Isao	Kimata	NEC Corporation	
Satoshi	Kinoshita	NEC Corporation	
Hiroki	Tagato	NEC Corporation	
Sergio	Lobo	NXP Semiconductors	
Gregory	Grisco	Oracle Corporation	
Jahangir	Nakra	Procter & Gamble	
Trong	Le	Psion Teklogix Inc.	
Craig	Harmon	Q.E.D. Systems	
Peter	Spreadborough	Reva Systems	
Sudhir	Hasbe	Samsys	
Stephan	Haller	SAP	
Steve	Winkler	SAP	
Sengu	Elango	Savi	
Neal	Herman	Savi	
Don	Ahn	Savi Technology	
L. Julia	Zhu	Savi Technology	
Pankaj	Shukla	Symbol	
Jong	Park	Tibco	
Keith	Rider	Tyco / ADT	
Bob	Sawdye	Tyco / ADT	
David	Harty	VeriSign	
Richard	Campero	Vue Technology	

3826

3827 The following list in corporate alphabetical order contains all companies that were
3828 opt'd-in to the Reader Operations Working Group and have signed the
3829 EPCglobal IP Policy.

3830

Company
(ETRI) Electronics and Telecommunications Research Institute
7iD (formerly EOSS GmbH)
Accenture
Acer Cybercenter Service Inc.
Altria Group, Inc./Kraft Foods
Applied Wireless (AWID)
Ark Tech Ltd
Auto-ID Labs - Cambridge
Auto-ID Labs - Japan
Auto-ID Labs - MIT
BEA Systems
Blackbay Ltd.
CAEN
Cisco
Convergence Sys Ltd
Dai Nippon Printing
Denso Wave Inc
ECO, Inc.
EPCglobal Inc.
FEIG Electronic
France Telecom
Fujitsu Ltd
GlobeRanger
GS1 Australia EAN
GS1 Germany (CCG)
GS1 Hong Kong
GS1 Japan
GS1 South Korea
GS1 Taiwan (EAN)

GS1 US
IBM
Impinj
Infineon Technologies NA Corp
Institute for Information Industry
Intellex
Intermec
Internet Initiative Japan, Inc.
Johnson & Johnson
Kimberly-Clark Corp
KL-NET
Korea Computer Servs, Ltd
LIT (Research Ctr for Logistics Info Tech)
Loftware, Inc.
Manhattan Associates
MET Labs
Metarights
Metro
Microelectronics Technology, Inc.
Mstar Semiconductor
NCR
NEC Corporation
NXP Semiconductors
OatSystems
ODIN Technologies
Omron
Oracle Corporation
Panda Int'l Transp Ltd
Pango Networks, Inc.
Paxar
PepsiCo
Polaris Networks
Procter & Gamble
Psion Teklogix Inc.
Q.E.D. Systems

Raining Data Corporation
RetailTech
Reva Systems
RFIP Ltd. (formerly Radio Freq Ident Ctr)
RFXCEL Corp
SAP
Savi Technology
Sirit
SOFTBANK TELECOM Corp. (Japan)
Supply Insight, Inc.
SyGade Solutions
Symbol Technologies, Inc.
T3C Incorporated
Tagent Corporation
TagSys
TEGO, Inc.
ThingMagic, LLC
Tibco
Toppan Printing Co
Toray International, Inc.
Tyco / ADT
Ussen Limited Company
VeriSign
Vocollect
Vue Technology
Wal-Mart
Wish Unity
Yuen Foong Yu Paper

3831

3832