

Reader Protocol Conformance Requirements Version 1.1



Reader Protocol (RP 1.1) Conformance Requirements V 1.1

Date: March 3, 2006

1	Conformance Test Framework.....	3
2	Conformance Tests Guidelines.....	4
3	Reader Protocol Objects and Methods.....	5
4	List of Reader Functionalities.....	9
5	Negative Tests.....	33
6	Test Case Requirements (TCR).....	39
6.1	Requirement command getEPC (TCR1).....	39
6.2	Requirement command setName / getName (TCR2).....	39
6.3	Requirement command setRole / getRole (TCR3).....	40
6.4	Requirement command getTimeTicks (TCR4).....	40
6.5	Requirement command getCurrentSource (TCR5).....	41
6.6	Requirement command getCurrentDataSelector (TCR6).....	41
6.7	Requirement command resetToDefaultSettings (TCR7).....	42
6.8	Requirement command reboot (TCR8).....	42
6.9	Requirement command goodbye (TCR9).....	43
6.10	Requirement command rawReadIDs (TCR10).....	43
6.11	Requirement command readIDs (TCR11).....	44
6.12	Requirement command kill (TCR12).....	44
6.13	Requirement command setReadTimeout (TCR12).....	45
6.14	Requirement command Trigger::getMaxNumberSupported (TCR13).....	45
6.15	Requirement command TagSelector::getMaxNumberSupported (TCR14).....	46
6.16	Requirement command EventType::getSupportedTypes (TCR15).....	46
6.17	Requirement command TriggerType::getSupportedTypes (TCR16).....	47
6.18	Requirement command FieldName::getSupportedNames (TCR17).....	47
6.19	Requirement command Source::getMaxNumberSupported (TCR18).....	48
6.20	Requirement command NotificationChannel::getMaxNumberSupported (TCR19).....	48
6.21	Requirement command DataSelector::getMaxNumberSupported (TCR20).....	49

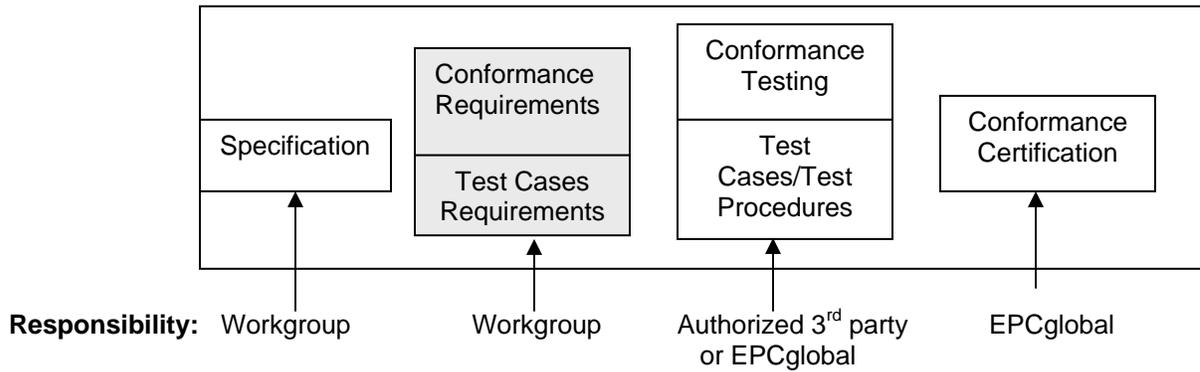
Reader Protocol Conformance Requirements

Version 1.1

6.22	Requirement command TagField::getMaxNumberSupported (TCR21)	49
6.23	Requirement command ReadPoint.getName (TCR22)	50
6.24	Requirement for a failed command behavior (TCR23)	50
6.25	Requirement for a unsupported handshake (TCR24)	51
6.26	Requirement for an invalid handshake format (TCR25)	51

Reader Protocol Conformance Requirements Version 1.1

1 Conformance Test Framework



Reader Protocol Conformance Requirements

Version 1.1

2 Conformance Tests Guidelines

1. Always use the specification as a baseline and keep it in synchronization with the Conformance Requirements
 - If you can find that a specification is not clear or wrong, update the specification first.
 - “SHALLS” are used to test for conformance and are considered mandatory requirements.
 - “SHOULDs” are considered to be recommendations or one of several possibilities and are not used to test for conformance.
 - “MAYs” are things that are permissible but are not to be used to test for conformance.
 - Avoid conditional uses and dependencies among conformance requirements
 - Even though the “SHALLS” are the primary reason for developing the bulk of material used to develop certification testing procedures, It’s a good idea to also develop test case requirements for the optional requirements captured in the 2.4 table. There are several reasons to develop the optional test cases too, as follows: 1) You may discover issues with the current specification and there is time to change the spec., and 2) You may decide to switch an Optional requirement to a Mandatory “SHALL.”
2. Develop Conformance Requirements in parallel to development of Specification.
3. Engage different editor(s) from the Workgroup then those writing the specification
4. For the record document anecdotal information. For example, regulatory agencies that have an impact on a particular part of the specification, if it is not already referenced in the specification.
5. Avoid describing specific implementations in Specifications and Conformance Requirements
 - Implementations can be very creative and may change from the time the Conformance Requirements Document is written. This might preclude some vendors and their products from becoming tested and certified.
6. Conduct a formal review of the whole Conformance Requirements Document
 - Even if test cases have been developed by outside 3rd party (as it was a case in Gen2)
7. “Applies to (ref)” Column of Tables 2.3 and 2.4
This column should not be filled in for Mandatory Requirements. For Optional Requirements, it is recommended that a WG fills this column.
8. Test Cases Requirements (TCRs)
 - Clearly define test input data, test operation sequences and expected results
 - Defined in such a detail that it can be independently implemented
 - Tests are defined to be atomic such that no test is dependent upon the execution of another test. This modularity allows tests to be developed independently and in parallel.
 - Test Case Requirements are developed by others into full test cases or test procedures.

Reader Protocol Conformance Requirements Version 1.1

3 Reader Protocol Objects and Methods

This section lists all the RP 1.1 objects and the methods applicable to each of the objects. This also states if support of the methods by a reader is mandatory (SHALL) or optional (MAY).

Object	Method	Conformance Requirement	Test Case
ReaderDevice	ReaderDevice.getEPC	SHALL	TCR1
	ReaderDevice.getManufacturer	MAY	
	ReaderDevice.getModel	MAY	
	ReaderDevice.getHandle	MAY	
	ReaderDevice.setHandle	MAY	
	ReaderDevice.getName	SHALL	TCR2
	ReaderDevice.setName	SHALL	TCR2
	ReaderDevice.getRole	SHALL	TCR3
	ReaderDevice.setRole	SHALL	TCR3
	ReaderDevice.getTimeTicks	SHALL	TCR4
	ReaderDevice.getTimeUTC	MAY	
	ReaderDevice.setTimeUTC	MAY	
	ReaderDevice.getManufacturerDescription	MAY	
	ReaderDevice.getCurrentSource	SHALL	TCR5
	ReaderDevice.setCurrentSource	MAY	
	ReaderDevice.getCurrentDataSelector	SHALL	TCR6
	ReaderDevice.setCurrentDataSelector	MAY	
	ReaderDevice.removeSources	MAY	
	ReaderDevice.removeAllSources	MAY	
	ReaderDevice.getSource	MAY	
	ReaderDevice.getAllSources	MAY	
	ReaderDevice.removeDataSelectors	MAY	
	ReaderDevice.removeAllDataSelectors	MAY	
	ReaderDevice.getDataSelector	MAY	
	ReaderDevice.getAllDataSelectors	MAY	
	ReaderDevice.removeNotificationChannels	MAY	
	ReaderDevice.removeAllNotificationChannels	MAY	
	ReaderDevice.getNotificationChannel	MAY	
	ReaderDevice.getAllNotificationChannels	MAY	
	ReaderDevice.removeTriggers	MAY	
	ReaderDevice.removeAllTriggers	MAY	
	ReaderDevice.getTrigger	MAY	
	ReaderDevice.getAllTriggers	MAY	
	ReaderDevice.removeTagSelectors	MAY	
	ReaderDevice.removeAllTagSelectors	MAY	
	ReaderDevice.getTagSelector	MAY	
	ReaderDevice.getAllTagSelectors	MAY	
	ReaderDevice.removeTagFields	MAY	

Reader Protocol Conformance Requirements

Version 1.1

	ReaderDevice.removeAllTagFields	MAY	
	ReaderDevice.getTagField	MAY	
	ReaderDevice.getAllTagFields	MAY	
	ReaderDevice.resetToDefaultSettings	SHALL	TCR7
	ReaderDevice.reboot	SHALL	TCR8
	ReaderDevice.goodbye	SHALL	TCR9
	ReaderDevice.getReadPoint	MAY	
	ReaderDevice.getAllReadPoints	MAY	
Source	Source.create	MAY	
	Source.getName	MAY	
	Source.isFixed	MAY	
	Source.addReadPoints	MAY	
	Source.removeReadPoints	MAY	
	Source.removeAllReadPoints	MAY	
	Source.getReadPoint	MAY	
	Source.getAllReadPoints	MAY	
	Source.addReadTriggers	MAY	
	Source.removeReadTriggers	MAY	
	Source.removeAllReadTriggers	MAY	
	Source.getReadTrigger	MAY	
	Source.getAllReadTriggers	MAY	
	Source.addTagSelectors	MAY	
	Source.removeTagSelectors	MAY	
	Source.removeAllTagSelectors	MAY	
	Source.getTagSelector	MAY	
	Source.getAllTagSelectors	MAY	
	Source.getGlimpsedTimeout	MAY	
	Source.setGlimpsedTimeout	MAY	
	Source.getObservedThreshold	MAY	
	Source.setObservedThreshold	MAY	
	Source.getObservedTimeout	MAY	
	Source.setObservedTimeout	MAY	
	Source.getLostTimeout	MAY	
	Source.setLostTimeout	MAY	
	Source.rawReadIDs	SHALL	TCR10
	Source.readIDs	SHALL	TCR11
	Source.read	MAY	
	Source.writeID	MAY	
	Source.write	MAY	
	Source.kill	SHALL	TCR12
	Source.getReadCyclesPerTrigger	MAY	
	Source.setReadCyclesPerTrigger	MAY	
	Source.getMaxReadDutyCycle	MAY	
	Source.setMaxReadDutyCycle	MAY	
	Source.getReadTimeout	MAY	
	Source.setReadTimeout	SHALL	TCR12
	Source.getSession	MAY	
	Source.setSession	MAY	
	Source.getMaxNumberSupported	SHALL	TCR18

Reader Protocol Conformance Requirements

Version 1.1

TagFieldValue	TagFieldValue.TagFieldName:String	MAY	
	TagFieldValue.Value:binary	MAY	
ReadPoint	ReadPoint.getName	SHALL	TCR22
Trigger	Trigger.create	MAY	
	Trigger.getMaxNumberSupported	SHALL	TCR13
	Trigger.getName	MAY	
	Trigger.getType	MAY	
	Trigger.getValue	MAY	
	Trigger.fire	MAY	
TagSelector	TagSelector.create	MAY	
	TagSelector.getMaxNumberSupported	SHALL	TCR14
	TagSelector.getName	MAY	
	TagSelector.getTagField	MAY	
	TagSelector.getValue	MAY	
	TagSelector.getMask	MAY	
	TagSelector.getInclusiveFlag	MAY	
NotificationChannel	NotificationChannel.create	MAY	
	NotificationChannel.getName	MAY	
	NotificationChannel.getAddress	MAY	
	NotificationChannel.getEffectiveAddress	MAY	
	NotificationChannel.setAddress	MAY	
	NotificationChannel.getDataSelector	MAY	
	NotificationChannel.setDataSelector	MAY	
	NotificationChannel.addSources	MAY	
	NotificationChannel.removeSources	MAY	
	NotificationChannel.removeAllSources	MAY	
	NotificationChannel.getSource	MAY	
	NotificationChannel.getAllSources	MAY	
	NotificationChannel.addNotificationTriggers	MAY	
	NotificationChannel.removeNotificationTriggers	MAY	
	NotificationChannel.removeAllNotificationTriggers	MAY	
	NotificationChannel.getNotificationTrigger	MAY	
	NotificationChannel.getAllNotificationTriggers	MAY	
	NotificationChannel.readQueuedData	MAY	
	NotificationChannel.getMaxNumberSupported	SHALL	TCR19
	DataSelector	DataSelector.create	MAY
DataSelector.getName		MAY	
DataSelector.addFieldNames		MAY	
DataSelector.removeFieldNames		MAY	
DataSelector.removeAllFieldNames		MAY	
DataSelector.getAllFieldNames		MAY	
DataSelector.addEventFilters		MAY	
DataSelector.removeEventFilters		MAY	
DataSelector.removeAllEventFilters		MAY	
DataSelector.getAllEventFilters		MAY	
DataSelector.addTagFieldNames	MAY		

Reader Protocol Conformance Requirements Version 1.1

	DataSelector.removeTagFieldNames	MAY	
	DataSelector.removeAllTagFieldNames	MAY	
	DataSelector.getAllTagFieldNames	MAY	
	DataSelector.getMaxNumberSupported	SHALL	TCR20
EventType	EventType.getSupportedTypes	SHALL	TCR15
TriggerType	TriggerType.getSupportedTypes	SHALL	TCR16
FieldName	FieldName.getSupportedNames	SHALL	TCR17
TagField	TagField.create	MAY	
	TagField.getName	MAY	
	TagField.getTagFieldName	MAY	
	TagField.setTagFieldName	MAY	
	TagField.getMemoryBank	MAY	
	TagField.setMemoryBank	MAY	
	TagField.getOffset	MAY	
	TagField.setOffset	MAY	
	TagField.getLength	MAY	
	TagField.setLength	MAY	
	TagField.getMaxNumberSupported	SHALL	TCR21

<i>Number of SHALL commands:</i>	25
<i>Number of MAY commands:</i>	126
<i>Total number of commands:</i>	151

Reader Protocol Conformance Requirements

Version 1.1

4 List of Reader Functionalities

This section lists all the SHALL reader functionalities in terms of protocol sub clause (section by section), requirements and if the clause is a Must (hard) or Optional (conditional) support for a reader.

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
1	4 Reader Layer	The Reader Protocol provides commands (which all conforming readers SHALL implement) through which Hosts can discover the capabilities of a particular reader.	hard	See Command List.
2	4.1.3 Data Acquisition Stage	In the case where a source corresponds to the direct reading of EPCglobal-compliant Class 1 RF Tags via an antenna, a read cycle SHALL correspond to a single protocol run.	conditional	By design
3	4.1.3 Data Acquisition Stage	ReadCyclesPerTrigger (count): The number of read cycles that SHALL be attempted, if the ReadTimeout does not expire first.	conditional	only readers supporting that command
4	4.1.3 Data Acquisition Stage	If a second or subsequent ReadTrigger is received before the complete series of read cycles for the first trigger has finished, then the series for the first trigger SHALL complete normally and then be followed by a new series of read cycles.	conditional	Only readers supporting triggers
6	4.2.1 Smoothing/Event Generation Stage	When generating events, individual tags SHALL be tracked separately, and the generated events SHALL indicate to what tag they pertain	conditional	Only readers supporting events
7	4.2.1 Smoothing/Event Generation Stage	It is clear from the above example that to do event generation, the Reader SHALL maintain some state information on a per-tag basis: in this case, the state information is whether the tag was seen in the previous read cycle or not.	conditional	Only readers supporting events
8	4.2.1 Smoothing/Event Generation Stage	When multiple sources are involved, this information SHALL also be maintained per-source, so that the state information and events are maintained for each unique combination of tag and source. The state information is collectively referred to as the <i>tag list</i> .	conditional	only readers supporting multiple sources
9	4.2.1 Smoothing/Event Generation Stage	Note that if cObserved_Threshold is equal to zero, then a tag that is seen for the first time goes from isUnknown to isGlimpsed and immediately to isObserved . In this case, simultaneous evNew, evGlimpsed and a evObserved events SHALL be generated.	conditional	Only readers supporting events
10	4.2.1 Smoothing/Event Generation Stage	Similarly, a tag in the isLost state that is seen again would move to isGlimpsed and immediately to isObserved. In this case, simultaneous evGlimpsed and a evObserved events SHALL be generated	conditional	Only readers supporting events
11	4.2.1 Smoothing/Event Generation Stage	If cLost_Timeout is zero, then a tag that moves to the isLost state goes directly to isUnknown. In this case, both a evLost and a evPurged event SHALL be generated at the same time.	conditional	Only readers supporting events
12	4.2.1 Smoothing/Event Generation Stage	Likewise, a reader that does not distinguish between isGlimpsed and isObserved states SHALL restrict cObserved_Threshold to be equal to zero, and restrict the available events to include evObserved but not evGlimpsed.	conditional	Only readers supporting events
13	4.3.2 Report Buffer Stage	When data has been delivered to the Host, the	conditional	Only readers supporting

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
		report buffer SHALL be cleared.		notifications
14	5 Object Model	There are no explicit operations on this object. A compliant Reader SHALL however have at least one command channel with a default address where it is listening to incoming commands. This default address SHALL be listed in the documentation of a compliant reader.	hard	Tested implicitly in all tests that send a command to the reader.
15	5.2 Sources	A Reader SHALL be able to report the number of Source objects it supports. A Source object that has been named and added to the ReaderDevice object is said to be allocated.	hard	TCR18
16	5.2 Sources	A pre-configured Source that cannot have its ReadPoint assignment modified (i.e., ReadPoints cannot be added or removed). Obviously, to be useful a fixed Source SHALL come with at least one ReadPoint assigned to it.	conditional	Only readers having preconfigured sources
17	5.2.1 Source Organization	A Reader SHALL be configured with at least one Source and a Reader MAY be configured to support multiple Sources.	hard	TCR5
18	5.2.1 Source Organization	When only a single Source is present, the operations requested through the Reader Protocol apply to that single Source without ambiguity. In this case, the Reader's Current Source SHALL always reference the single Source object.	conditional	Only readers with a single, fixed source
19	5.2.1 Source Organization	When multiple Sources are available, the Reader SHALL provide the means to select what Sources SHALL be used to acquire data at any given time and the Reader MAY provide the means of applying different configuration parameters (e.g., ReadTriggers, TagSelectors, etc.) to the different Sources.	conditional	Only readers supporting multiple sources. They SHALL implement the addSources() etc. commands.
20	5.2.1 Source Organization	A Reader SHALL support at least one Source. To minimize Reader requirements, a single Source can be supported that comes from the manufacturer pre-configured and fixed (i.e., its ReadPoint assignments cannot be dynamically re-configured by a host).	hard	Same as 25: TCR5
21	5.2.1 Source Organization	Each Source SHALL have a unique, but otherwise arbitrary identifier (i.e., name).	conditional	Only readers supporting multiple sources
22	5.2.1 Source Organization	Reader commands that take a Source input parameter SHALL use the Current Source when the Source input parameter is omitted.	hard for SHALL commands	<i>There are no reader commands that take an optional Source parameter!</i>
23	5.2.1 Source Organization	Readers SHALL be configured in one of the topologies outlined here: Zero ReadPoints, One ReadPoint Multiple ReadPoints	hard	No need for testing.
24	5.2.1 Source Organization	A Reader with just one ReadPoint SHALL have just one Source that is pre-configured to be allocated with a pre-defined name and with the one ReadPoint assigned to it.	semi-conditional	see 32
25	5.2.1 Source Organization	A Reader with just one Source SHALL come pre-configured with this Source already allocated with a pre-defined name.	semi-conditional	see 32
26	5.3 Triggers	Continuous: The activity is triggered as rapidly as possible. For ReadTriggers, regulatory limits on duty cycles SHALL be taken into account. For NotificationTriggers, this means a Notification SHALL be sent whenever a new Event appears in the Report Buffer.	conditional	Only readers supporting triggers

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
27	5.3 Triggers	IO Edge: The documentation for a conforming reader SHALL specify to what connectors or other physical elements each port and pin setting correspond.	conditional	Only readers supporting triggers
28	5.3 Triggers	<VendorName>: <Any> Vendors can specify proprietary triggering mechanisms. The name of these triggers SHALL have the vendor name and a colon as prefix.	conditional	Only readers supporting triggers and having vendor-specific triggers
29	5.4 TagSelectors	In the filter mask M, all bit positions where the value is important for the filtering shall be set to 1.		This is an instruction for the user of the protocol!
30	5.4.1 Multiple Tag Selectors	Multiple TagSelector objects can be associated with any given source. Each of these filter objects is specified to be either <i>inclusive</i> (meaning that only tags matching the filter SHALL be reported) or <i>exclusive</i> (meaning that a tag SHALL only be reported if it does not match the filter).	conditional	Only readers supporting filtering
31	5.4.1 Multiple Tag Selectors	If multiple TagSelectors are used, a tag SHALL be reported only if the following two conditions hold: The tag matches at least one of the inclusive patterns; <i>and</i> The tag does not match any of the exclusive patterns.	conditional	Only readers supporting filtering
32	5.4.1 Multiple Tag Selectors	As a special case, if zero inclusive patterns are defined, the first check SHALL be omitted.	conditional	Only readers supporting filtering
33	5.5 Events	<VendorName>: <Any> Arbitrary vendor-specified event. The name of these events SHALL have the vendor name and a colon as prefix.	conditional	Only readers having vendor-specific events
34	5.5 Events	A conforming Reader that implements notifications SHALL generate at least one of either two event types: evGlimpsed or evNew.	conditional	Only readers implementing notifications
35	5.6 Channels	The Reader Protocol defines a handshake protocol (see section 10.1) that a Host and Reader can use to establish what binding shall be used for communications.		
36	5.6 Channels	A CommandChannel connection SHALL always be initiated by a Host.	hard	No test necessary.
37	5.6 Channels	Reader SHALL contain at least one CommandChannel	hard	see 21
38	5.6 Channels	All supported CommandChannels SHALL be pre-configured (e.g., the address to which a host can connect) and allocated to the ReaderDevice object by the Reader manufacturer.	hard	Check documentation if at least one command channel address is defined. See also 21.
39	5.6 Channels	NotificationChannels SHALL be created by a Host via a CommandChannel request.	hard	No test necessary.
40	5.6 Channels	This request (NotificationChannel.create) SHALL specify a connection address, which includes the specification (see section 9.1 for a description of the address URI-style address syntax) of the transport to be used.	conditional	Only readers supporting notifications
41	5.7 DataSelectors	If a DataSelector has some fields for which there is no information or that are inapplicable (e.g., a trigger name when no triggers are used), these fields SHALL be ignored and not included in the output.	conditional	

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
42	5.7 DataSelectors	ReaderEPC SHALL The ReaderEPC attribute of the reader.	hard	Implicitly covered in TCR11
43	5.7 DataSelectors	ReaderName SHALL The Reader Name attribute of the reader.	hard	Implicitly covered in TCR11
44	5.7 DataSelectors	ReaderRole SHALL The Reader Role attribute of the reader.	hard	Implicitly covered in TCR11
45	5.7 DataSelectors	TagID SHALL The ID of the tag in raw binary format.	hard	Implicitly covered in TCR11
46	5.7 DataSelectors	TagIDasPureURI If the tag ID is not a valid EPC, then the raw URI form SHALL be used, e.g urn:epc:raw:64.20018283527919	conditional	Only if that fieldName is supported
47	5.7 DataSelectors	AllSupported SHALL All fields that are supported by this reader. Which set of fields this is, is vendor-dependent. This is also the set of fields that SHALL be reported when no DataSelector is defined.	hard	Implicitly covered in TCR11
48	5.7 DataSelectors	<Vendorname>:<Any> MAY Vendors MAY define additional fields. The name of these fields SHALL have the vendor name and a colon as prefix.	conditional	Only if there are vendor-specific field names
49	5.7 DataSelectors	In addition, TagFieldName objects can be added to a data selector to have specific memory fields on the tag read and reported. If several TagFieldName objects are added, all these fields SHALL be read.	conditional	Only readers supporting user memory and TagFieldNames
50	5.9 Object Names	Each object SHALL be identified by an arbitrary, but unique name, settable by the host. A name is unique per object-type, meaning that there SHALL NOT be any two objects of the same type with the same name, but there MAY be two objects of different types with the same name. The latter is not advisable, however.	conditional	Only readers implementing create methods.
51	6 Reader Layer-Commands	The maximum length of settable string attributes is vendor-dependent, but all readers SHALL support a length of at least 64 bytes. Attempts to set a string of excessive length SHALL result in an error.	conditional	
52	6 Reader Layer-Commands	All commands are <i>atomic</i> , meaning that they SHALL be executed either completely or not at all.	hard	By design – no test possible
53	6 Reader Layer-Commands	If there is an error while trying to execute a command, the state of the reader including all attribute values of all objects SHALL remain unchanged. For example, when adding arrays of things (e.g., FieldNames to a DataSelector), and one value is not supported or not known, then no values SHALL be added and an error SHALL be raised.	hard	By design – no test possible For last sentence: TCR 23
54	6 Reader Layer-Commands	If the command cannot be executed correctly, the return value SHALL be undefined and an error condition SHALL be raised.	hard	The return value cannot be tested by definition, since it is undefined. Second SHALL: TCR 23
55	6 Reader Layer-Commands	All strings SHALL be capable of being represented with UTF-8 encoding.	hard	By design
56	6.1 Object Reader Device	The ReaderDevice object represents the Reader. It is a singleton object that SHALL exist by default. There are therefore no commands to create or delete it.	hard	Tested implicitly in all tests using the ReaderDevice object.

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
57	6.1 Object Reader	The ReaderDevice object keeps lists of all Sources, DataSelectors, NotificationChannels, Triggers and TagSelectors known. The objects SHALL be added to these lists implicitly when created.	conditional	Only readers supporting these objects.
58	6.1.1 ReaderDevice.getEPC	Returns the EPC code of the Reader. Since the EPC is set by the manufacturer, there is no setEPC() command. Compliance Requirement: Compliant systems SHALL implement this command	hard	TCR1
59	6.1.4 ReaderDevice.getHandle	Data Type: integer. The handle of the Reader. If no handle has been set, 0 (zero) SHALL be returned.	conditional	
60	6.1.6 ReaderDevice.getName	Compliant systems SHALL implement this command.	hard	TCR2
61	6.1.6 ReaderDevice.getName	Data Type: string. The name of the reader. If no name has been set, an empty string SHALL be returned.	hard	TCR 2
62	6.1.7 ReaderDevice.setName	Compliant systems SHALL implement this command.	hard	TCR2
63	6.1.8 ReaderDevice.getRole	Compliant systems SHALL implement this command.	hard	TCR3
64	6.1.9 ReaderDevice.setRole	Compliant systems SHALL implement this command.	hard	TCR3
65	6.1.10 ReaderDevice.getTimeTicks	Compliant systems SHALL implement this command.	hard	TCR4
66	6.1.13 ReaderDevice.getManufacturerDescription	The format is implementation-dependent and SHALL be described in the reader documentation.	conditional	Only if reader supports this command
67	6.1.14 ReaderDevice.getCurrentSource	Returns the current Source of the Reader. If the Reader supports one or more Sources then the Reader SHALL provide a Current Source.	hard	TCR5
68	6.1.14 ReaderDevice.getCurrentSource	The Current Source MAY be pre-configured by the Reader manufacturer or the Reader MAY support commands that allow setting the Current Source. If the Reader supports only one source, then the Current Source SHALL be a constant equated to the one source.	hard	see 29, 37
69	6.1.14 ReaderDevice.getCurrentSource	Compliant systems SHALL implement this command.	hard	TCR5
70	6.1.16 ReaderDevice.getCurrentDataSelector	Per default, this SHALL be a DataSelector with only the FieldName 'AllSupportedFields' added, i.e, all supported fields and all supported event types will be reported.	hard	see 62
71	6.1.16 ReaderDevice.getCurrentDataSelector	Compliant systems SHALL implement this command.	hard	TCR6

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
72	6.1.18 ReaderDevice.removeSources	If one or more of the Sources given are not known, or if some of the Sources to be removed are currently not associated with this Reader, these are ignored and all other Sources SHALL be removed and the command SHALL complete successfully. However, if one or more Sources of the list is a fixed Source, the command SHALL return in an error and no Sources SHALL be removed.	conditional	Only if reader supports this command
73	6.1.21 ReaderDevice.getAllSources	Returns all Sources currently associated with this Reader. If no Sources are currently associated with the Reader, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
74	6.1.22 ReaderDevice.removeDataSelectors	If one or more of the DataSelectors given are not known, or if some of the DataSelectors to be removed are currently not associated with this Reader, these are ignored and all other DataSelectors SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
75	6.1.25 ReaderDevice.getAllDataSelectors	Returns all DataSelectors currently associated with the Reader. If no DataSelectors are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
76	6.1.26 ReaderDevice.removeNotificationChannels	If one or more of the NotificationChannels given are not known, or if some of the NotificationChannels to be removed are currently not associated with the Reader, these are ignored and all other NotificationChannels SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
77	6.1.29 ReaderDevice.getAllNotificationChannels	Returns all NotificationChannels currently associated with the Reader. If no NotificationChannels are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
78	6.1.30 ReaderDevice.removeTriggers	If one or more of the Triggers given are not known, or if some of the Triggers to be removed are currently not associated with the Reader, these are ignored and all other Triggers SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
79	6.1.33 ReaderDevice.getAllTriggers	Returns all Triggers currently associated with the Reader. If no Triggers are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
80	6.1.34 ReaderDevice.removeTagSelectors	If one or more of the TagSelectors given are not known, or if some of the TagSelectors to be removed are currently not associated with the Reader, these are ignored and all other TagSelectors SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
81	6.1.37 ReaderDevice.getAllTagSelectors	If no TagSelectors are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
82	6.1.38 ReaderDevice.removeTagFields	If one or more of the TagFields given are not known, or if some of the TagFields to be removed are currently not associated with the Reader, these are ignored and all other TagFields SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
83	6.1.41 ReaderDevice.getAllTagFields	If no TagFields are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
84	6.1.42 ReaderDevice.getAllTagFields	The documentation of the Reader SHALL provide a definition what the default settings are.	conditional	Only if reader supports this command
85	6.1.42 ReaderDevice.getAllTagFields	Compliant systems MAY implement this command.	conditional:	Only if reader supports this command
86	6.1.43 ReaderDevice.reboot	This command SHALL behave as follows: Check if a reboot is possible. If so the command SHALL return without an error, else an error is returned.	hard	TCR8
87	6.1.43 ReaderDevice.reboot	Compliant systems SHALL implement this command.	hard	TCR8
88	6.1.44 ReaderDevice.goodbye	Compliant systems SHALL implement this command.	hard	TCR9
89	6.2.1 Source.create	The Source SHALL implicitly be added to the list of all Sources kept by the ReaderDevice object.	conditional	Only if reader supports this command
90	6.2.4 Source.addReadPoints	If one or more of the ReadPoints given are not known, an error SHALL be raised and no ReadPoints SHALL be added. If some of the ReadPoints to be added are already associated with this Source, only the not yet associated ReadPoints shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
91	6.2.4 Source.addReadPoints	If the Reader implementation does not allow ReadPoints to be shared between Sources and any of the specified ReadPoints are already assigned to another Source, then this command SHALL report the error ERROR_READPOINT_IN_USE and no ReadPoints will be added to the Source.	conditional	Only if reader supports this command
92	6.2.5 Source.removeReadPoints	If one or more of the ReadPoints given are not known, or if some of the ReadPoints to be removed are currently not associated with this Source, these are ignored and all other ReadPoints SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
93	6.2.9 Source.addReadTriggers	If one or more of the Triggers given are not supported or are not known, an error SHALL be raised and no Triggers SHALL be added. If some of the Triggers to be added are already associated with this Source, only the not yet associated Triggers shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
94	6.2.10 Source.removeReadTriggers	If one or more of the Triggers given are not supported or not known, or if some of the Triggers to be removed are currently not associated with this Source, these are ignored and all other Triggers SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
95	6.2.13 Source.getAllReadTriggers	If no Triggers are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
96	6.2.14 Source.addTagSelectors	If one or more of the TagSelectors given are not known, an error SHALL be raised and no TagSelectors SHALL be added. If some of the TagSelectors to be added are already associated with this Source, only the not yet associated TagSelectors shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
97	6.2.14 Source.addTagSelectors	If some of the TagSelectors to be added are already associated with this Source, only the not yet associated TagSelectors shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
98	6.2.15 Source.removeTagSelectors	If one or more of the TagSelectors given are not known, or if some of the TagSelectors to be removed are currently not associated with this Source, these are ignored and all other TagSelectors SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
99	6.2.18 Source.getAllTagSelectors	If no TagSelectors are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	
100	6.2.20 Source.setGlimpsedTimeout	The value of the glimpsedTimeout attribute, measured in milliseconds. It SHALL be equal to or greater than 0.	conditional	Only if reader supports this command
101	6.2.22 Source.setObservedThreshold	threshold - Data Type: integer. The value of the observedThreshold attribute, measured in milliseconds. It SHALL be equal to or greater than 0.	conditional	Only if reader supports this command
102	6.2.24 Source.setObservedTimeout	The value of the observedTimeout attribute, measured in milliseconds. It SHALL be equal to or greater than 0.	conditional	Only if reader supports this command
103	6.2.26 Source.setLostTimeout	The value of the lostTimeout attribute, measured in milliseconds. It SHALL be equal to or greater than 0.	conditional	Only if reader supports this command
104	6.2.27 Source.rawReadIDs	The resulting ReadReport SHALL be formatted according to the given DataSelector	hard	Implicit in TCR11
105	6.2.27 Source.rawReadIDs	Implementations MAY impose restrictions on the usage of this command if Read Triggers are active. Such restrictions SHALL be documented.	hard	Check documentation.
106	6.2.27 Source.rawReadIDs	Compliant systems SHALL implement this command.	hard	TCR10
107	6.2.27 Source.rawReadIDs	If no DataSelector parameter is given, the Reader's current DataSelector SHALL be used.	hard	TCR10
108	6.2.28 Source.ReadIDs	The number of read cycles performed SHALL be determined by the value of the Source attribute ReadCyclesPerTrigger.	conditional	See also 4
109	6.2.28 Source.ReadIDs	The resulting ReadReport is formatted according to the given DataSelector. If a tag is seen in several read cycles, it SHALL only be reported once.	hard	TCR11
110	6.2.28 Source.ReadIDs	Implementations MAY impose restrictions on the usage of this command if Read Triggers are active. Such restrictions SHALL be documented.	hard	Check documentation.
111	6.2.28 Source.ReadIDs	Compliant systems SHALL implement this command.	hard	TCR11
112	6.2.28 Source.ReadIDs	If no DataSelector parameter is given, the Reader's current DataSelector SHALL be used.	hard	TCR11

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
113	6.2.29 Source.read	The number of read cycles performed SHALL be determined by the value of the Source attribute ReadCyclesPerTrigger.	conditional	Only if reader supports this command
114	6.2.29 Source.read	If a tag is seen in several read cycles, it SHALL only be reported once.	conditional	Only if reader supports this command
115	6.2.29 Source.read	Such use of passwords and their application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
116	6.2.29 Source.read	Implementations MAY impose restrictions on the usage of this command if Read Triggers are active. Such restrictions SHALL be documented.	conditional	Only if reader supports this command
117	6.2.29 Source.read	If no DataSelector parameter is given, the Reader's current DataSelector SHALL be used.	conditional	Only if reader supports this command
118	6.2.30 Source.writeID	The Source object's list of associated TagSelectors, if any, SHALL be ignored during the execution of this command.	conditional	Only if reader supports this command
119	6.2.30 Source.writeID	The second password in the parameter list can correspond to a tag's EPC <i>kill password</i> . Such use of passwords and their application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
120	6.2.30 Source.writeID	Implementations MAY impose restrictions on the usage of this command if Read Triggers are active. Such restrictions SHALL be documented.	conditional	Only if reader supports this command
121	6.2.31 Source.write	The Source object's list of associated TagSelectors, if any, SHALL be ignored during the execution of this command.	conditional	Only if reader supports this command
122	6.2.31 Source.write	Such use of passwords and their application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
123	6.2.31 Source.write	Implementations MAY impose restrictions on the usage of this command if Read Triggers are active. Such restrictions SHALL be documented.	conditional	Only if reader supports this command
124	6.2.32 Source.kill	A tag's RF protocol may require one or more passwords (or <i>lock code</i>) to successfully complete this operation. Such use of passwords and their application to specific RF protocols SHALL be documented.	hard	Check documentation.
125	6.2.32 Source.kill	Compliant systems SHALL implement this command.	hard	TCR12
126	6.2.34 Source.setMaxReadDutyCycle	cycles - Data Type: integer. The value of the readCyclesPerTrigger attribute. It SHALL be greater than 0.	conditional	Only if reader supports this command
127	6.2.38 Source.getReadTimeout	Compliance Requirement: Compliant systems SHALL implement this command.	conditional	Only if reader supports this command
128	6.2.38 Source.getReadTimeout	timeout - Data Type: integer. The value of the ReadTimeout attribute, measured in milliseconds. It SHALL be equal to or greater than 0.	conditional	Only if reader supports this command
129	6.2.39 Source.getSession	The application of this attribute is dependent upon a tag's RF protocol. The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
130	6.2.40 Source.setSession	The application of this attribute is dependent upon a tag's RF protocol. The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
131	6.3.1 TagFieldValue.TagFieldName:String	The TagField object SHALL be associated with the corresponding Source object. The TagField object specifies the tag memory location for the corresponding data field.	conditional	Only if reader supports this command

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
132	6.4.1 ReadPoint.getName	Compliance Requirement: Compliant systems SHALL implement this command.	hard	TCR 22
133	6.5 Object Trigger	Source and NotificationChannel objects SHALL contain zero Trigger objects by default.	conditional	Only if these objects are supported.
134	6.5 Object Trigger	A Trigger SHALL be activated as soon as it is added to another object. It SHALL be deactivated once it has been removed from all objects it had been added to, i.e. when it is not associated with any other object.	conditional	Only if triggers are supported
135	6.5 Object Trigger	type=<rising falling>;port=<n>;pin=<n>IOEdge The trigger fires when the specified i/o pin on the specified port changes state. The type SHALL be 'rising' or 'falling'.	conditional	Only if triggers are supported
136	6.5 Object Trigger	vendor=<name>.... The <name> is required and SHALL contain the vendor's name.	conditional	Only if triggers are supported
137	6.5.1 Trigger.create	The Trigger object's type and value establish the conditions whereby the trigger SHALL <i>fire</i> .	conditional	Only if reader supports this command
138	6.5.1 Trigger.create	If the Reader doesn't support the requested TriggerType, the error condition ERROR_PARAMETER_NOT_SUPPORTED shall be raised.	conditional	Only if reader supports this command
139	6.5.1 Trigger.create	The Trigger SHALL implicitly be added to the list of all Triggers kept by the ReaderDevice object.	conditional	Only if reader supports this command
140	6.5.2 Trigger.getMaxNumberSupported	Compliance Requirement: Compliant systems SHALL implement this command.	hard	TCR13
141	6.6 Object TagSelector	A Source object SHALL contain zero TagSelector objects by default.	conditional	Only if filtering is supported.
142	6.6.1 TagSelector.create	The TagSelector SHALL implicitly be added to the list of all TagSelectors kept by the ReaderDevice object.	conditional	Only if reader supports this command
143	6.6.2 TagSelector.getMaxNumberSupported	Compliance Requirement: Compliant systems SHALL implement this command.	hard	TCR14
144	6.7 Object CommandChannel	There are no explicit operations on this object. A compliant Reader SHALL however have at least one command channel with a default address where it is listening to incoming commands. This default address SHALL be listed in the documentation of a compliant reader	hard	see 21
145	6.8 Object NotificationChannel	Only events coming from these Sources SHALL be reported. It also has a DataSelector that determines what data should be reported to the host. If no DataSelector is explicitly associated, then the Reader's <i>current</i> DataSelector SHALL be used. Notification SHALL be sent whenever the associated Trigger fires. Alternatively, a host MAY also query the NotificationChannel for its contents, using the command NotificationChannel.readQueuedData().	conditional	Only if notifications and these objects are supported
146	6.8.1 NotificationChannel.create	The NotificationChannel SHALL implicitly be added to the list of all NotificationChannels kept by the ReaderDevice object.	conditional	Only if reader supports this command

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
147	6.8.5 NotificationChannel.setAddress	If <i>connect mode</i> was specified, then the return value does not apply and the Reader SHALL return zero. Otherwise, if listen mode was specified, then the return SHALL return the <i>port number</i> assigned by the Reader to listen for host Notification Channel connections.	conditional	Only if reader supports this command
148	6.8.8 NotificationChannel.addSources	If one or more of the Sources given are not known, an error SHALL be raised and no Sources SHALL be added. If some of the Sources to be added are already associated with this NotificationChannel, only the not yet associated Sources shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
149	6.8.9 NotificationChannel.removeSources	If one or more of the Sources given are not known, or if some of the Sources to be removed are currently not associated with this NotificationChannel, these are ignored and all other Sources SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
150	6.8.12 NotificationChannel.getAllSources	If no Sources are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
151	6.8.13 NotificationChannel.addNotificationTriggers	If one or more of the Triggers given are not supported or are not known, an error SHALL be raised and no Triggers SHALL be added. If some of the Triggers to be added are already associated with this NotificationChannel, only the not yet associated Triggers shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
152	6.8.14 NotificationChannel.removeNotificationTriggers	If one or more of the Triggers given are not supported or not known, or if some of the Triggers to be removed are currently not associated with this NotificationChannel, these are ignored and all other Triggers SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
153	6.8.17 NotificationChannel.getAllNotificationTriggers	If no Triggers are currently associated with this object, the command SHALL complete successfully and an empty list SHALL be returned.	conditional	Only if reader supports this command
154	6.8.18 NotificationChannel.readQueuedData	Implementations MAY impose restrictions on the usage of this command if Notification Triggers are active. Such restrictions SHALL be documented.	conditional	Only if reader supports this command
155	6.8.18 NotificationChannel.readQueuedData	An optional flag to indicate if the report buffer should be cleared after the ReadReport is returned. The default SHALL be true.	conditional	Only if reader supports this command
156	6.9 Object DataSelector	DataSelector objects are used to define what data SHALL be reported in notification messages or by commands that take a DataSelector object as parameter. Only events and data fields that are added to the respective lists SHALL be reported.	hard	Implicitly in TCR11
157	6.9.1 DataSelector.create	The DataSelector SHALL implicitly be added to the list of all DataSelectors kept by the ReaderDevice object.	conditional	Only if reader supports this command
158	6.9.3 DataSelector.addFieldNames	If one or more of the FieldNames given are not supported, an error SHALL be raised and no FieldNames SHALL be added. If some of the FieldNames to be added are already associated with this DataSelector, only the not yet associated FieldNames shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
159	6.9.3 DataSelector.addFieldNames	fields – Data Type: FieldName[]. The list of FieldNames to be added. These data fields SHALL from now on be reported.	conditional	Only if reader supports this command

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
160	6.9.4 DataSelector.removeFieldNames	If one or more of the FieldNames given are not supported, these are ignored and all other FieldNames SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
161	6.9.7 DataSelector.addEventFilters	If one or more of the EventTypes given are not supported, an error SHALL be raised and no EventTypes SHALL be added. If some of the EventTypes to be added are already associated with this DataSelector, only the not yet associated EventTypes shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
162	6.9.7 DataSelector.addEventFilters	Only events of the types that have been added SHALL be reported. That is, the EventTypes added are to be used as <i>inclusive</i> filters.	conditional	Only if reader supports this command
163	6.9.7 DataSelector.addEventFilters	The list of EventTypes to be added. Events of these types SHALL from now on be reported.	conditional	Only if reader supports this command
164	6.9.8 DataSelector.removeEventFilter	If one or more of the EventTypes given are not supported, or if some of the EventTypes to be removed are currently not associated with this DataSelector, these are ignored and all other EventTypes SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
165	6.9.11 DataSelector.addTagFieldNames	If some of the given names to be added are already associated with this DataSelector, only the not yet associated names shall be added and the command SHALL complete successfully.	conditional	Only if reader supports this command
166	6.9.11 DataSelector.addTagFieldNames	fieldNames – Data Type: string[]. The list of TagField names to be added. These data fields SHALL from now on be reported.	conditional	Only if reader supports this command
167	6.9.12 DataSelector.removeTagFieldNames	If one or more of the names given are not currently associated with this DataSelector, these are ignored and all other names SHALL be removed and the command SHALL complete successfully.	conditional	Only if reader supports this command
168	6.10.1.1 EventType.getSupportedTypes	Compliance Requirement: Compliant systems SHALL implement this command.	hard	TCR15
169	6.10.2.1 TriggerType.getSupportedTypes	Compliance Requirement: Compliant systems SHALL implement this command.	hard	TCR16
170	6.10.3.1 FieldName.getSupportedNames	Compliance Requirement: Compliant systems SHALL implement this command.	hard	TCR17
171	6.11.1 TagField.create	The TagField SHALL implicitly be added to the list of all TagFields kept by the ReaderDevice object.	conditional	Only if reader supports this command
172	6.11.2 TagField.getName	A TagField object's name SHALL be unique whereas a TagField object's TagFieldName attribute MAY be shared (see 6.11.3 TagField.getTagFieldName for details).	conditional	Only if reader supports this command
173	6.11.5 TagField.getMemoryBank	The application of this attribute is dependent upon a tag's RF protocol. The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
174	6.11.6 TagField.setMemoryBank	The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
175	6.11.7 TagField.getOffset	The exact application of this attribute is dependent upon a tag's RF protocol. However, the offset SHALL always be specified in units of memory <i>bits</i> . The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
176	6.11.8 TagField.setOffset	The exact application of this attribute is dependent upon a tag's RF protocol. However, the offset SHALL always be specified in units of memory <i>bits</i> . The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
177	6.11.9 TagField.getLength	The exact application of this attribute is dependent upon a tag's RF protocol. However, the length SHALL always be specified in units of memory <i>bits</i> . The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
178	6.11.10 TagField.setLength	The exact application of this attribute is dependent upon a tag's RF protocol. However, the length SHALL always be specified in units of memory <i>bits</i> . The use of this attribute and its application to specific RF protocols SHALL be documented.	conditional	Only if reader supports this command
179	7.1.2 Vendor-defined Notification	Vendor-specific notifications SHALL be marked as such and include the vendor name to disambiguate notifications from different vendors.	conditional	Only if reader supports this command
180	7.2.1 Goodbye	Data Type: void. This notification SHALL take no parameters.	conditional	Only if reader supports this command
181	8.3 Command Errors	SHALL: The error code and/or error name. At least one of these fields SHALL be included in the Error structure. The exact contents depend on the binding, and SHALL be constant at the binding level, e.g. MTB X can be defined as always including the error code (but not the error name)	hard	TCR 23
182	8.3 Command Errors	SHALL (where applicable): Information about the error cause. For example, for input parameter errors, information on which parameter failed needs to be included. Depending on the binding, this can be either the name or the index of the parameter. Vendor extensions are allowed here.	conditional	
183	8.3 Command Errors	SHALL (where applicable): A vendor name or identifier for the responses to vendor-specific commands SHALL be given	conditional	Only vendor specific errors.
184	8.4 Vendor Extensions	Vendors SHALL use the error conditions defined above for the commands of the standard command set.	conditional	Only vendor specific errors.
185	8.4 Vendor Extensions	These vendor-defined error conditions SHALL use names starting with <VENDOR>_ERROR...	conditional	Only vendor specific errors.
186	9 Message/Transport Bindings (MTBs)	1. The handshake exchange that is used to negotiate the specific Message Layer encoding. This exchange occurs in a text-based format that SHALL be understood by all implementations, regardless of Reader Layer encoding support.	hard	Tested implicitly
187	10.1.3.1 Initiation	Upon establishing a connection via the transport mechanism, the Sender SHALL send the Sender-Handshake. Upon establishing a connection via the transport mechanism, the Receiver SHALL read the Sender-Handshake and SHALL then write the Receiver-Handshake back to the Sender with values indicating the response to the requested support.	hard	Tested implicitly

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
188	10.1.3.2 Versioning and Capabilities Request	Note that this does not necessarily mean that the Receiver is actually operating at the requested level; it just means that it is able to respond properly to commands, and that all required functionality and commands (i.e., those termed "SHALL") are supported.		Just a citational "SHALL"!
189	10.1.3.2 Versioning and Capabilities Request	In the case of a Receiver operating at a version different than the one indicated in the request, and the Receiver deciding that it is not able to operate at the required level, the Receiver SHALL respond with the value of "NO".	hard	TCR 24
190	10.1.3.3 Extensibility	In Reader Protocol 1.1, this field SHALL be set to all 0's when sent and SHALL be ignored upon receipt.	hard	TCR 25
191	10.2.1 Transport Address Notation	In accordance with section 9.1. <i>Address Notation</i> , the URI naming convention SHALL be used to address a serial communications transport.	conditional	MTB-specific
192	10.2.2 Serial Interface Physical Characteristics	A Reader SHALL come from a manufacturer with published default physical device characteristics including, but not limited to, the following items: Baud rate Data bits Stop bits Parity Flow control The Host SHALL initially communicate on the Reader's serial interface using the published default device characteristic values.	conditional	MTB-specific
193	10.2.3 Human-to-Machine Mode (TTY)	While in TTY mode, the Reader SHALL maintain the following state machine.	conditional	MTB-specific
194	10.2.3 Human-to-Machine Mode (TTY)	Quiet The Reader can only report asynchronous data over the Notification Channel when the Reader is in the Quiet state. As it begins sending notification data, the Reader SHALL transition to the Notification Output state. From the Quiet state, a null command (i.e., a command terminator) SHALL cause the Reader to transition to the Notification Stopped state. Whenever the Reader transitions to the Quiet state (including the initial entry), the Reader SHALL send a command prompt to the user. 1- The Reader SHALL enter the Quiet state initially. 2- If it begins sending notification data, the Reader SHALL transition to the Notification Output state.	conditional	MTB-specific

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
		<p>5- If a null command is detected the Reader SHALL transition to the Notification Stopped state.</p> <p>7- The Reader SHALL transition to the Command Input state whenever input is detected from the user other than a command terminator.</p>		
195	10.2.3 Human-to-Machine Mode (TTY)	<p style="text-align: center;">Command Input</p> <p>The Reader SHALL enter the Command Input state whenever a single input character is received from the user other than a command terminator. Once in the Command Input state, the Reader SHALL stay in this state until a command terminator is received or a timeout occurs.</p> <p>8- When a command terminator is received the Reader SHALL transition to the Command Processing state.</p> <p>10-The Reader SHALL transition from the Command Input state back to the Quiet state if the command terminator is not received within a specified timeout period.</p>	conditional	MTB-specific
196	10.2.3 Human-to-Machine Mode (TTY)	<p style="text-align: center;">Command Processing</p> <p>The Reader SHALL enter the Command Processing state from the Command Input state once a command terminator is received. The Reader SHALL stay in the Command Processing state until the last character of the command response has been transmitted to the user. The Reader SHALL then transition to the Quiet State.</p>	conditional	MTB-specific
197	10.2.3 Human-to-Machine Mode (TTY)	<p style="text-align: center;">Notification Output</p> <p>The Reader SHALL enter the Notification Output state from the Quiet state when it has asynchronous data to transmit on the Notification Channel. Once in the Notification Output state, the Reader SHALL ignore any user input received. Once notification data has completed, the Reader SHALL re-enter the Quiet state.</p> <p>It is possible that the Reader MAY generate asynchronous notification data continuously. In this case the user MAY become locked out and unable to enter a command. To manage this situation, while in the Notification Output state, the Reader SHALL detect a null command.</p> <p>3- When a null command is detected the Reader SHALL interrupt the notification output at a reasonable breaking point (implementation dependent) and transition to the Notification Stopped state.</p>	conditional	MTB-specific

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
		4- When notification data has completed, the Reader SHALL re-enter the Quiet state.		
198	10.2.3 Human-to-Machine Mode (TTY)	<p>Notification Stopped</p> <p>The Reader SHALL transition to the Notification Stopped state from either the Quiet state or the Notification Output state whenever a null command is detected. While in the Notification Stopped state, the Reader SHALL hold all notification data until the Reader re-enters the Quiet state.</p> <p>The Reader SHALL transition from the Notification Stopped state to the Command Input state whenever a single input character is received other than a command terminator. While in the Notification Stopped state, if a command terminator is received, the Reader SHALL remain in the Notification Stopped state.</p> <p>6- When a single input character is received other than a command terminator the Reader SHALL transition to the Command Input state.</p> <p>11- Once in the Notification Stopped state for thirty (30) seconds or longer, the Reader SHALL transition to the Notification Output state where it will resume sending queued notification information.</p>	conditional	MTB-specific
199	10.2.3 Human-to-Machine Mode (TTY)	If the handshake is not performed, then the Reader SHALL use pre-defined encoding and message framing rules established for its serial TTY binding.	conditional	MTB-specific
200	10.2.3.1.1 Control Channel	<p>When a user wants to establish a TTY serial connection with a Reader, the user MAY send the handshake request (see section 10.1Error! Reference source not found.) or the user SHALL send a connection initialization message which consists of the ASCII text:</p> <p>TTY<CommandTerminator></p>	conditional	MTB-specific

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
201	10.2.3.1.1 Control Channel	When a Reader receives a TTY connection request, it SHALL accept or reject the request by sending the appropriate Receiver-Handshake response that SHALL be framed as follows: <handshake reply><EOLTerminator>	conditional	MTB-specific
202	10.2.3.1.1 Control Channel	When a Reader accepts a TTY request on the Control Channel, the Reader SHALL refuse subsequent requests for connection on that port as long as the TTY connection between Reader and Host exists on that port.	conditional	MTB-specific
203	10.2.3.1.2 Notification Channel	The Reader SHALL send notification information following the rules established by the state machine described in section 10.2.3. The Reader SHALL wait for the Control Channel to be quiet before transmitting Notification Channel data. In this state, the Reader can send asynchronous messages to the user. All Notification Channel messages SHALL be prefixed with the text: NOTIFICATION:<EOLTerminator>	conditional	MTB-specific
204	10.2.3.1.2 Notification Channel	Any user input detected while a notification message is being written SHALL be discarded by the Reader (see the state machine description in section 10.2.3.1.1). The user SHALL be obliged to re-enter discarded data.	conditional	MTB-specific
205	10.2.3.2.1 Control Channel	If the Reader terminates the Control Channel it SHALL send a termination notification to the user: <EOLTerminator> (see section 10.2.3.1.1) TERMINATED<EOLTerminator>	conditional	MTB-specific
206	10.2.4 Machine-to-Machine Mode M2M	The serial M2M mode transport layer requires that all messages sent by either the Host or the Reader SHALL be framed as follows: <Header><message content><Trailer>	conditional	MTB-specific

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
207	10.2.4 Machine-to-Machine Mode M2M	The Reader documentation SHALL include information about the port(s) a reader is listening to by default for incoming requests. During a connection handshake initiated by the Host, the Host SHALL establish a new, unique port number. If the connection is established by the Reader, the Host SHALL first issue a NotificationChannel.setAddress command which defines a unique port number to be used for the corresponding channel. When the Reader connects it, SHALL use the port number established by the setAddress command.	conditional	MTB-specific
208	10.2.4 Machine-to-Machine Mode M2M	The message identifier is always established by the sender. The receiver SHALL acknowledge a message received using the sender's message identifier. If the sender resends a message (i.e., no ACK is received), then the sender SHALL reuse the same message identifier every time the message is resent until the message is acknowledged.	conditional	MTB-specific
209	10.2.4 Machine-to-Machine Mode M2M	All messages sent by either the Host or the Reader SHALL be acknowledged by the receiver with the following text: <Length>A<Port-Number><Message-ID><Checksum>	conditional	MTB-specific
210	10.2.4 Machine-to-Machine Mode M2M	The ACK message header includes the channel and message identifiers that SHALL match the sender's identifiers. The sender SHALL wait for this ACK before proceeding. The sender MAY have a configurable timeout that establishes the maximum time it waits for an ACK. The receiver MAY also have a configurable timeout that establishes the maximum time the receiver SHALL wait to receive the end of a message.	conditional	MTB-specific
211	10.2.4 Machine-to-Machine Mode M2M	If the sender times out waiting for an ACK, the sender SHALL re-send the message. When re-sending a message, the sender SHALL reuse the same message identifier every time the message is resent until the message is acknowledged.	conditional	MTB-specific
212	10.2.4.1 Connection Establishment (Connect)	The Reader SHALL listen for an incoming serial request from the Host (Control Channel) on a pre-configured serial I/O communications port. A handshake protocol SHALL be used to establish serial connections.	conditional	MTB-specific
213	10.2.4.1.1 Control Channel	When a Host wants to establish a M2M serial connection with a Reader, the Host SHALL send a connection initialization message (handshake) defined for the in section 10.1.2.1. The Reader SHALL respond with the corresponding handshake response message (also defined in section 10.1.2.2).	conditional	MTB-specific
214	10.2.4.1.1 Control Channel	When a Reader receives a serial connection request, it SHALL accept or reject the request by sending the appropriate Receiver-Handshake with the appropriate responses as defined in Section 10.1.2.	conditional	MTB-specific

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
215	10.2.4.1.1 Control Channel	<p>As specified in section 10.2.4 Error! Reference source not found., these messages SHALL be encapsulated within a header and trailer. And, each message SHALL be acknowledged in addition to the appropriate message response. i.e.:</p> <p>Host: <Header><serial handshake request><Trailer></p> <p>Reader: ACK</p> <p>Reader: <Header><serial handshake reply><Trailer></p> <p>Host: ACK</p>	conditional	MTB-specific
216	10.2.4.1.2 Notification Channel	The Host SHALL listen for an incoming serial request from the Reader (Notification Channel) on a pre-configured serial I/O communications port.	conditional	MTB-specific
217	10.2.4.1.2 Notification Channel	Host and the Reader SHALL use this port number in the header of any subsequent messages exchanged on the notification channel established.	conditional	MTB-specific
218	10.2.4.2.1 Control Channel	The format is implementation encoding dependent. If the Reader terminates the Control Channel it SHALL send a termination notification to the Host as specified in the reader command reference. If the termination is indicated through notification then no further communication SHALL be expected from the reader after the notification record.	conditional	MTB-specific
219	10.3.1 Transport Address Notation	<p>In accordance with section 9.1 Error! Reference source not found. <i>Address Notation</i>, the URI naming convention SHALL be used to address a TCP communications transport. The <i>transport</i> of the URI is the keyword:</p> <p>tcp</p> <p>The locator of the URI has the syntax:</p> <p><host address>[:<logical port number>]</p>	conditional	MTB-specific
220	10.3.2 Connection Establishment (Connect)	A Reader SHALL listen for an incoming TCP request from a Host (Control Channel) on the RP_PORT. Other ports MAY be supported but such support is outside the scope of this specification. Also, if commanded by a Host, a Reader MAY listen for an incoming TCP request from a Host to establish a Notification Channel connection. In this case the Reader SHALL assign a port for listening.	conditional	MTB-specific
221	10.3.2 Connection Establishment (Connect)	The default value for RP_PORT shall be registered with IANA (Internet Assigned Numbers Authority) and shall be reflected in this specification when obtained.	conditional	MTB-specific
222	10.3.2.1 Control Channel	When a Reader receives a TCP connection request, it SHALL accept or reject the request by sending the appropriate Receiver-Handshake with the appropriate responses as defined in	conditional	MTB-specific

Reader Protocol Conformance Requirements Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
		Section 10.1.2.		
223	10.3.2.2 Notification Channel	In connect mode the Reader SHALL initiate a connection to the Host. In listen mode the Reader SHALL listen for a connection initiated by the Host.	conditional	MTB-specific
224	10.3.2.2 Notification Channel	When a Host receives a TCP connection request, it SHALL accept or reject the request by sending the appropriate Receiver-Handshake with the appropriate responses as defined in Section 10.1.2.	conditional	MTB-specific
225	10.3.2.2 Notification Channel	When a Reader receives a TCP connection request, it SHALL accept or reject the request by sending the appropriate Receiver-Handshake with the appropriate responses as defined in Section 10.1.2.	conditional	MTB-specific
226	10.3.3.1 Control Channel	If the termination is indicated through notification then no further communication SHALL be expected from the reader after the notification record.	conditional	MTB-specific
227	10.4 HTTP	If Readers and Hosts support HTTPS, they SHALL use HTTP Over TLS as defined in RFC 2818.	conditional	MTB-specific
228	10.4.1 Transport Address Notation	In accordance with section 9.1 <i>Address Notation</i> , the URI naming convention SHALL be used to address a serial communications transport. The <i>transport</i> of the URI is the keyword: http The locator of the URI has the syntax: <host address>[:<logical port number>]	conditional	MTB-specific
229	10.4.2 Connection Establishment (Connect)	All HTTP messages SHALL obey the following rules: HTTP 1.1 is required All standard RP headers named 'RP-*' HTTP Header 'Content-Type' is required HTTP Header 'Content-Length' is required Additional HTTP headers MAY be present	conditional	MTB-specific
230	10.4.2 Connection Establishment (Connect)	All HTTP Requests SHALL also obey the following rules: Default URL Method = POST HTTP Header 'RP-Sender-Signature' is optional	conditional	MTB-specific

Reader Protocol Conformance Requirements Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
		<p>HTTP Header 'RP-Spec-Version' is optional</p> <p>HTTP Header 'RP-Response-Content-Type' is optional</p> <p>HTTP Header 'RP-Response-ACKNAK' is optional</p>		
231	10.4.2 Connection Establishment (Connect)	<p>All HTTP Responses SHALL also obey the following rules:</p> <p>Status Code is required</p> <p>HTTP Header 'RP-Receiver-Signature' is optional</p> <p>HTTP Header 'RP-Spec-Version-OK' is optional</p> <p>HTTP Header 'RP-Request-Content-Type-OK' is optional</p> <p>HTTP Header 'RP-Response-Content-Type-OK' is optional</p> <p>HTTP Header 'RP-Response-ACKNAK-OK' is optional</p>	conditional	MTB-specific
232	10.4.2.3 HTTP Header 'Content-Type' is required	<p>The Content-Type header field SHALL always be present, as per the message format in the body of the request/response:</p> <p style="text-align: center;">Content-Type: application/octet-stream</p> <p>Binary Message Format, as per handshake <sender-format> = 'B1' Content-Type: text/plain; charset=utf-8</p> <p>Text Message Format, as per handshake <sender-format> = 'T1' Content-Type: text/xml; charset=utf-8</p> <p>XML Message Format, as per handshake <sender-format> = 'X1'</p>	conditional	MTB-specific
233	10.4.2.4 HTTP 'Content-Length' is required	<p>The Content-Length header field SHALL always be present, as per HTTP 1.1. See RFC2616 for more details.</p>	conditional	MTB-specific
234	10.4.2.6.1 Default URL	<p>Readers and Hosts SHALL, by default, support the URL</p> <p>Control Channel = http://server:port/control where server is the Reader's address and port the TCP/IP port on which the reader is listening, as per standard HTTP rules. The default port number SHALL be noted in the documentation of the Reader. Other resources may be supported but such support is outside the scope of this specification.</p>	conditional	MTB-specific

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
235	10.4.2.6.6 HTTP Header 'RP-Response-Content-Type' is optional	If 'no', the response's body SHALL be empty (i.e. Content-Length: 0). <i>NOTE:</i> The HTTP Response is always required; this header merely controls whether or not the response has a body (it always has a header).	conditional	MTB-specific
236	10.4.2.7.1 Status Code is required	All HTTP responses SHALL contain a status code, as per "6.1.1. Status Code and Reason Phrase" and "10. Status Code Definitions" in the HTTP 1.1 standard [RFC2616]. The status code is used to indicate communicate or handshake errors; command set errors SHALL NOT use the status code to convey this information. For example, if an unsupported command is received by a reader the HTTP Response SHALL contain a status code of 200 (OK) with the body containing a standard error response containing an error code of ERROR_COMMAND_NOT_SUPPORTED, formatted as per the appropriate message format.	conditional	MTB-specific
237	10.4.2.8 Control Channel	When a Reader receives an HTTP request, it SHALL accept or reject the request by sending an HTTP Response with the appropriate structure (headers, body, etc.) as defined by this transport.	conditional	MTB-specific
238	10.4.2.9 Notification Channel	When a Host receives an HTTP request, it SHALL accept or reject the request by sending an HTTP Response with the appropriate structure (headers, body, etc.) as defined by this transport.	conditional	MTB-specific
239	11.1 XML Message Format	The naming convention of all element tag names and attribute names follow the EPCglobal XML Design guidelines. The names SHALL use the Lower Camel Case ('LCC') convention, i.e. the leading character in the first word SHALL be in lower case and the leading characters of each subsequent word in upper case.	conditional	MTB-specific
240	11.1.2 Data Types	A compliant reader SHALL uniformly represent the EPC datatype using the "raw hex" URI as defined in the EPCglobal Tag Data Standards.... The hexadecimal encoding shall be case insensitive. The "raw hex" URI format specifies the length of the EPC data field.	conditional	MTB-specific
241	11.1.3 Read Report encoding	A read report SHALL be arranged in one of the two ways: <ol style="list-style-type: none"> 1. A list of tags with no source information. 2. List of sub reports with one sub report per source. Each source report contains a list of tags associated with that source. It MAY also have vendor defined data specific to source. Type 1 SHALL be used when no source info is requested in the DataSelector, else type 2 SHALL be used.	conditional	MTB-specific
242	11.1.4.1 Message Format	Every command message is encoded within the <command> element. The command message body SHALL contain a message id. For non-static methods, it SHALL also contain a <i>target name</i> , identifying the object instance to which the command is applied.	conditional	MTB-specific
243	11.1.4.1 Message Format	All input parameters are encoded as child elements of the command in the following manner:	conditional	MTB-specific

Reader Protocol Conformance Requirements Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
		<p><parametername>parametervalue</parametername></p> <p>For example, the command</p> <p>NotificationChannel. setAddress(addr: address): void</p> <p>invoked as</p> <p>channel1.setAddress("http://myhost.com:8080/EPCApp")</p> <p>SHALL be formatted as <command></p> <p><id>1234</id></p> <p><targetName>channel1</targetName></p> <p><notificationChannel></p> <p><setAddress></p> <p><addr>http://myhost.com:8080/EPCApp</addr></p> <p></setAddress></p> <p></notificationChannel></p> <p></command></p>		
244	11.1.4.2 Vendor Extensions	Vendor defined commands SHALL appear under an element in their own namespace which is different from EPC namespace, within <command> element.	conditional	MTB-specific
245	11.1.4.2 Vendor Extensions	<p>SHALL be formatted as <command></p> <p><id>1234</id></p> <p><targetName>myString</targetName></p> <p><acme:string xmlns:acme="http://www.acme.com/rp/extension"></p> <p>></p> <p><acme:concatenate></p> <p><acme:left>abc</acme:left></p> <p><acme:right>xyz</acme:right></p> <p></acme:concatenate></p> <p></acme:string></p> <p></command></p>	conditional	MTB-specific
246	11.1.5.1 Message Format	1. The <id> value SHALL exactly match the value in the corresponding <command>	conditional	MTB-specific
247	11.1.5.1 Message Format	4. Error information SHALL be encoded in an <error> node.	conditional	MTB-specific

Reader Protocol Conformance Requirements

Version 1.1

No.	Protocol SubClause	Requirements (Requirements, Command, ...)	hard / conditional	Condition / Testcase / Remarks
248	1.1.6 Notification XML message encoding (Reader-To-Host)	<p>Every notification message is encoded within the <notification> element. The notification message body SHALL contain a notification id. Optionally, the following information MAY also be added:</p> <p style="text-align: center;">Information about the Reader, like its EPC or its name</p> <p style="text-align: center;">The (notification) trigger that lead to this notification</p> <p style="text-align: center;">The name of the notification channel on which this notification is sent</p>	conditional	MTB-specific
249	1.1.6 Notification XML message encoding (Reader-To-Host)	<p>The content of a notification SHALL be encoded as one of the following child elements:</p> <ol style="list-style-type: none"> 1. ReadReport – encoded as <readReport> element. 2. goodbye – encoded as <goodBye> element. This element has no content inside. 	conditional	MTB-specific
250	11.1.6.1 Ack/NACK	Every ACK/NACK message is encoded within the <ackNack> element. The notification message body SHALL contain a notification id and a boolean field isAck which is true if the message is an ACK.	conditional	MTB-specific
251	11.2.1 External Encoding	The Text formatted message character set SHALL be in accordance with ISO 10646-1 (www.iso.org) which defines the Universal Character Set (UCS) Basic Multilingual Plane (BMP) or ISO 10646-2 (a newer addition that adds another plane to the character set). Note that ISO 10646-1:2000 (2 nd edition) is officially synchronized with the Unicode 3.0 standard (www.unicode.org/book/u2.html).	conditional	MTB-specific
252	11.2.1 External Encoding	The external encoding (i.e., the bits transmitted over a transport) rules for Text formatted messages SHALL adhere to the UCS Transformation Format, 8-bit form (UTF-8). This standard is defined in RFC 3629 and in Annex D of ISO 10646-1:2000.	conditional	MTB-specific
253	11.2.3 Data Types	<p><i>string</i>.</p> <p>A compliant reader SHALL uniformly represent the EPC datatype using the “raw hex” URI as defined in the EPCglobal Tag Data Standards.... The hexadecimal encoding shall be case insensitive. The “raw hex” URI format specifies the length of the EPC data field.</p>	conditional	MTB-specific

Reader Protocol Conformance Requirements Version 1.1

5 Negative Tests

Test ID	Test Class/Category	Spec. Reference	Test Purpose	Remarks
RP301	Mandatory/Handshake	10.1.2.1	This test case verifies the behavior of a handshake where the host requests an unsupported property of the reader.	
RP302	Mandatory/Handshake	10.1.2.1	This test case verifies the behavior of an undefined string sent instead of a valid handshake string.	
RP303	Optional/Read	6.2.29	This test case verifies the behavior of command <Source.read> without Source.	
RP304	Optional/Read	6.2.29	This test case verifies the behavior of command <Source.read> without a Source ReadPoint.	
RP305	Optional/Read	6.2.29	This test case verifies the behavior of command <Source.read> with the source using unknown ReadPoint.	
RP306	Optional/Read	6.2.29	This test case verifies the behavior of command <Source.read> with the source using unknown DataSelector as parameter.	
RP307	Optional/Read	6.2.29	This test case verifies the behavior of command <Source.read> with the source using unknown TagSelector as parameter.	
RP308	Optional/Read	6.2.29	This test case verifies the behavior of command <Source.read> after goodbye; and without handshaking.	
RP309	Optional/Read	6.2.29	This test case verifies the behavior of command <Source.read> after reboot; and without handshaking.	
RP310	Optional/Write	6.2.31	This test case verifies the behavior of command <Source.write> with invalid Password as parameter.	
RP311	Optional/Write	6.2.31	This test case verifies the behavior of command <Source.write> with unknown TagSelector as parameter.	
RP312	Mandatory/Kill	6.2.32	This test case verifies the behavior of command <Source.kill> with unknown Password as parameter.	
RP313	Mandatory/Kill	6.2.32	This test case verifies the behavior of command <Source.kill> with unknown TagSelector as parameter.	
RP314	Mandatory/Read	6.2.27	This test case verifies the behavior of command <Source.rawRead> with unknown DataSelector as parameter.	
RP315	Optional/Write	6.2.31	This test case verifies the behavior of command <Source.write> with unknown TagFieldValue as parameter.	
RP316	Optional/Write	6.2.30	This test case verifies the behavior of command <Source.rawWrite> with invalid TagFieldValue as parameter.	

Reader Protocol Conformance Requirements

Version 1.1

Test ID	Test Class/Category	Spec. Reference	Test Purpose	Remarks
RP317	Optional/Read	6.2.29	This test case verifies the behavior of command <NotificationChannel.create> with wrong address.	
RP318	Optional/Management	6.1.5	This test case verifies the command <ReaderDevice.SetHandle> with Negative value of Handle as parameter.	
RP319	Optional/Management	6.1.12	This test case verifies the command <ReaderDevice.SetTimeUTC> with setting the UTC as parameter to a negative value.	
RP320	Optional/Management	6.1.15	This test case verifies the command <ReaderDevice.SetCurrentSource> with setting the current source to an unknown source.	
RP321	Optional/Management	6.1.17	This test case verifies the command <ReaderDevice.SetCurrentDataSelector> with setting the current dataselector to an unknown data selector.	
RP322	Optional/Management	6.1.18	This test case verifies the command <ReaderDevice.removeSources > without any Source.	
RP323	Optional/Management	6.1.19	This test case verifies the command <ReaderDevice.removeAllSources> without any Source.	
RP324	Optional/Management	6.1.22	This test case verifies the command <ReaderDevice.removeDataSelector> without any DataSelector.	
RP325	Optional/Management	6.1.23	This test case verifies the command <ReaderDevice.removeAllDataSeltors> without any DataSelector.	
RP326	Optional/Management	6.1.24	This test case verifies the command <ReaderDevice.getDataSelector> with an unknown DataSelector name as a parameter.	
RP327	Optional/Management	6.1.26	This test case verifies the command <ReaderDevice.removeNotificationChannels> without any NotificationChannel.	
RP328	Optional/Management	6.1.27	This test case verifies the command <ReaderDevice.removeAllNotificationChannels> without any NotificationChannel.	
RP329	Optional/Management	6.1.28	This test case verifies the command <ReaderDevice.getNotificationChannel> with an unknown NotificationChannel name as a parameter.	
RP330	Optional/Management	6.1.30	This test case verifies the command <ReaderDevice.removeTriggers> without any Trigger.	
RP331	Optional/Management	6.1.31	This test case verifies the command <ReaderDevice.removeAllTriggers> without any Trigger.	
RP332	Optional/Management	6.1.32	This test case verifies the command <ReaderDevice.getTrigger> with an unknown Trigger name as a parameter.	
RP333	Optional/Management	6.1.34	This test case verifies the command <ReaderDevice.removeTagSelectors> without any TagSelector.	

Reader Protocol Conformance Requirements

Version 1.1

Test ID	Test Class/Category	Spec. Reference	Test Purpose	Remarks
RP334	Optional/Management	6.1.35	This test case verifies the command <code><ReaderDevice.removeAllTagSelectors></code> without any <code>TagSelector</code> .	
RP335	Optional/Management	6.1.36	This test case verifies the command <code><ReaderDevice.getTagSelector></code> with an unknown <code>TagSelector</code> name as a parameter.	
RP336	Optional/Management	6.1.38	This test case verifies the command <code><ReaderDevice.removeTagFields ></code> without any <code>TagField</code> .	
RP337	Optional/Management	6.1.39	This test case verifies the command <code><ReaderDevice.removeAllTagFields></code> without any <code>TagField</code> .	
RP338	Optional/Management	6.1.40	This test case verifies the command <code><ReaderDevice.getTagField></code> with an unknown <code>TagField</code> name as a parameter.	
RP339	Optional/Management	6.1.45	This test case verifies the command <code><ReaderDevice.getReadPoint></code> with an unknown <code>ReadPoint</code> name as a parameter.	
RP340	Optional/Management	6.2.4	This test case verifies the command <code><Source.addReadPoints></code> If the Reader implementation does not allow <code>ReadPoints</code> to be shared between <code>Sources</code> and any of the specified <code>ReadPoints</code> are already assigned to another <code>Source</code> , then this command SHALL report the error <code>ERROR_READPOINT_IN_USE</code>	
RP341	Optional/Management	6.2.5	This test case verifies the command <code><Source.removeReadPoints></code> without any <code>ReadPoint</code> .	
RP342	Optional/Management	6.2.6	This test case verifies the command <code><Source.removeAllReadPoints></code> without any <code>ReadPoint</code> .	
RP343	Optional/Management	6.2.7	This test case verifies the command <code><Source.getReadPoint></code> with an unknown <code>ReadPoint</code> name as a parameter.	
RP344	Optional/Management	6.2.10	This test case verifies the command <code><Source.removeReadTriggers></code> without any <code>ReadTrigger</code> .	
RP345	Optional/Management	6.2.11	This test case verifies the command <code><Source.removeAllReadTriggers></code> without any <code>ReadTrigger</code> .	
RP346	Optional/Management	6.2.12	This test case verifies the command <code><Source.getReadTrigger></code> with an unknown <code>Trigger</code> name as a parameter.	
RP347	Optional/Management	6.2.15	This test case verifies the command <code><Source.removeTagSelectors></code> without any <code>TagSelector</code> .	
RP348	Optional/Management	6.2.16	This test case verifies the command <code><Source.removeAllTagSelectors></code> without any <code>TagSelector</code> .	
RP349	Optional/Management	6.2.17	This test case verifies the command <code><Source.getTagSelector></code> with an unknown <code>TagSelector</code> name as a parameter.	
RP350	Optional/Management	6.2.20	This test case verifies the command <code><Source.setGlimpsedTimeout></code> with setting the parameter	

Reader Protocol Conformance Requirements

Version 1.1

Test ID	Test Class/Category	Spec. Reference	Test Purpose	Remarks
			timeout to a negative value.	
RP351	Optional/Management	6.2.22	This test case verifies the command <Source.setObservedThreshold> with setting the parameter threshold to a negative value.	
RP352	Optional/Management	6.2.24	This test case verifies the command <Source.setObservedTimeout> with setting the parameter timeout to a negative value.	
RP353	Optional/Management	6.2.26	This test case verifies the command <Source.setLostTimeout> with setting the parameter timeout to a negative value.	
RP354	Mandatory/Read	6.2.27	This test case verifies the command <Source.rawReadIDs> on a Source with no Readpoint assigned.	
RP355	Mandatory/Read	6.2.28	This test case verifies the command <Source.ReadIDs> on a Source with no Readpoint assigned.	
RP356	Optional/Read	6.2.29	This test case verifies the command <Source.ReadID> on a Source with no Readpoint assigned.	
RP357	Optional/Write	6.2.30	This test case verifies the command <Source.writeID> with no tags in the field of the Reader.	
RP358	Optional/Write	6.2.30	This test case verifies the command <Source.writeID> on a Source with no ReadPoint assigned.	
RP359	Optional/Write	6.2.31	This test case verifies the command <Source.write> with no tags in the field of the Reader.	
RP360	Optional/Write	6.2.31	This test case verifies the command <Source.write> on a Source with no ReadPoint assigned.	
RP361	Mandatory/Kill	6.2.32	This test case verifies the command <Source.Kill> with no tags in the field of the Reader.	
RP362	Mandatory/Kill	6.2.32	This test case verifies the command <Source.Kill> on a Source with no ReadPoint assigned.	
RP363	Optional/Management	6.2.34	This test case verifies the command <Source.setreadCyclesPerTrigger> with setting the parameter cycles to a negative value.	
RP364	Optional/Management	6.2.36	This test case verifies the command <Source.MaxReadDutyCycle> with setting the parameter dutyCycle to a value NOT between 0 and 100.	
RP365	Mandatory/Management	6.2.38	This test case verifies the command <Source.setReadTimeOut> with setting the parameter timeout to a negative value.	
RP366	Optional/Management	6.2.40	This test case verifies the command <Source.setSession> with setting the parameter session to a negative value.	

Reader Protocol Conformance Requirements

Version 1.1

Test ID	Test Class/Category	Spec. Reference	Test Purpose	Remarks
RP367	Optional/Management	6.5.1	This test case verifies the command <Trigger.create>with setting the parameter TriggerType to an unsupported value.	
RP368	Optional/Management	6.8.1	This test case verifies the command <NotificationChannel.create> with setting the parameter name to an already existing name.	
RP369	Optional/Management	6.8.5	This test case verifies the command <NotificationChannel.setAddress> with setting the parameter addr to an unknown value.	
RP370	Optional/Management	6.8.7	This test case verifies the command <NotificationChannel.setDataSelector> with setting the parameter dataSelector to an unknown value.	
RP371	Optional/Management	6.8.8	This test case verifies the command <NotificationChannel.addSources> with setting the parameter sources to an unknown Source name.	
RP372	Optional/Management	6.8.8	This test case verifies the command <NotificationChannel.addSources> with setting the parameter sources to an already existing Source name.	
RP373	Optional/Management	6.8.9	This test case verifies the command <NotificationChannel.removeSources> without any Source.	
RP374	Optional/Management	6.8.10	This test case verifies the command <NotificationChannel.removeAllSources> without any Source.	
RP375	Optional/Management	6.8.13	This test case verifies the command <NotificationChannel.addNotificationTriggers> with setting the parameter triggers to an unknown triggers name.	
RP376	Optional/Management	6.8.13	This test case verifies the command <NotificationChannel.addNotificationTriggers> with setting the parameter triggers to an already existing triggers name.	
RP377	Optional/Management	6.8.14	This test case verifies the command <NotificationChannel.removeNotificationTriggers> without any NotificationTrigger.	
RP378	Optional/Management	6.8.15	This test case verifies the command <NotificationChannel.removeAllNotificationTriggers> without any NotificationTrigger.	
RP379	Optional/Management	6.8.16	This test case verifies the command <NotificationChannel.getNotificationTrigger> with an unknown Trigger name as a parameter.	
RP380	Optional/Management	6.9.3	This test case verifies the command <DataSelector.addFieldNames> with setting the parameter fields to an unknown field name.	
RP381	Optional/Management	6.9.3	This test case verifies the command <DataSelector.addFieldNames> with setting the parameter fields to an already existing field name.	
RP382	Optional/Management	6.9.4	This test case verifies the command <DataSelector.removeFieldNames> without any FieldNames.	
RP383	Optional/Management	6.9.5	This test case verifies the command <DataSelector.removeAllFieldNames> without any FieldNames.	

Reader Protocol Conformance Requirements

Version 1.1

Test ID	Test Class/Category	Spec. Reference	Test Purpose	Remarks
RP384	Optional/Management	6.9.8	This test case verifies the command <DataSelector.removeEventFilters> without any EventFilters .	
RP385	Optional/Management	6.9.9	This test case verifies the command <DataSelector.removeAllEventFilters> without any EventFilters .	
RP386	Optional/Management	6.9.11	This test case verifies the command <DataSelector.addTagFieldNames> with setting the parameter fieldNames to an unknown field name.	
RP387	Optional/Management	6.9.11	This test case verifies the command <DataSelector.addTagFieldNames> with setting the parameter fieldNames to an already existing field name.	
RP388	Optional/Management	6.9.12	This test case verifies the command <DataSelector.removeTagFieldNames> without any TagFieldNames .	
RP389	Optional/Management	6.9.13	This test case verifies the command <DataSelector.removeAllTagFieldNames> without any TagFieldNames .	
RP390	Optional/Management	6.11.1	This test case verifies the command <TagField.create> with setting the parameter TagField name to an already existing name.	
RP391	Optional/Management	6.11.6	This test case verifies the command <TagField.setMemoryBank> with setting the parameter memoryBank to a negative value.	
RP392	Optional/Management	6.11.8	This test case verifies the command <TagField.setOffset> with setting the parameter offset to an unsupported value.	
RP393	Optional/Management	6.11.10	This test case verifies the command <TagField.setLength> with setting the parameter length to a negative value.	

Reader Protocol Conformance Requirements Version 1.1

6 Test Case Requirements (TCR)

6.1 Requirement command getEPC (TCR1)

TPIId: TCR-1		
Test Purpose: This test case verifies the implementation of the getEPC command. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>ReaderDevice.getEPC()</code>	The reader should return its EPC.

6.2 Requirement command setName / getName (TCR2)

TPIId: TCR-2		
Test Purpose: This test case verifies the implementation of the setName/getName command. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>String name = ReaderDevice.getName()</code> <code>ASSERT(name == "")</code>	The reader should return no error. The name should contain an empty string.
4	Call command: <code>ReaderDevice.setName("TestReader")</code>	The reader should return no error.
5	Call command: <code>String name = ReaderDevice.getName()</code>	The name string should contain the value "TestReader".

Reader Protocol Conformance Requirements Version 1.1

6.3 Requirement command setRole / getRole (TCR3)

TPIId: TCR-3		
Test Purpose: This test case verifies the implementation of the setRole/getRole command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>ReaderDevice.setRole("TestRole")</code>	The reader should return no error.
3	Call command: <code>String role = ReaderDevice.getRole()</code>	The role string should contain the vale "TestRole".

6.4 Requirement command getTimeTicks (TCR4)

TPIId: TCR-4		
Test Purpose: This test case verifies the implementation of the getTimeTicks command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>int tick1 = ReaderDevice.getTimeTicks()</code>	The int tick1 should contain a time ticks in msec of the readers uptime.
3	Wait one second	
4	Call command: <code>int tick2 = ReaderDevice.getTimeTicks()</code>	The int tick2 should contain a time tick in msec of the readers uptime which is one second greater than tick1.

Reader Protocol Conformance Requirements Version 1.1

6.5 Requirement command `getCurrentSource` (TCR5)

TPId: TCR-5		
Test Purpose: This test case verifies the implementation of the <code>getCurrentSource</code> command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: Source <code>s1 = ReaderDevice.getCurrentSource()</code>	The variable <code>s1</code> should contain a valid source object.

6.6 Requirement command `getCurrentDataSelector` (TCR6)

TPId: TCR-6		
Test Purpose: This test case verifies the implementation of the <code>getCurrentDataSelector</code> command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: DataSelector <code>ds1 = ReaderDevice.getCurrentDataSelector()</code>	The variable <code>ds1</code> should contain a valid data selector object.

Reader Protocol Conformance Requirements Version 1.1

6.7 Requirement command resetToDefaultSettings (TCR7)

TPId: TCR-7		
Test Purpose: This test case verifies the implementation of the resetToDefaultSettings command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>String role1 = ReaderDevice.getRole()</code>	The reader should return no error.
3	Call command: <code>ReaderDevice.setRole("XYZ")</code>	The reader should return no error.
4	Call command: <code>ReaderDevice.resetToDefaultSettings()</code>	The reader should return no error.
5	Call command: <code>String role2 = ReaderDevice.getRole()</code>	The reader should return no error.
6	Verify: <code>Role1 == role2</code>	The string role1 should match string role2.

6.8 Requirement command reboot (TCR8)

TPId: TCR-8		
Test Purpose: This test case verifies the implementation of the reboot command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>ReaderDevice.reboot()</code>	The reader should perform a full boot sequence and should then return to a mode where the reader can be connected again.

Reader Protocol Conformance Requirements Version 1.1

6.9 Requirement command goodbye (TCR9)

TPId: TCR-9		
Test Purpose: This test case verifies the implementation of the goodbye command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • None 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>ReaderDevice.goodbye()</code>	The reader should return no error and then the reader should close the connection.

6.10 Requirement command rawReadIDs (TCR10)

TPId: TCR-10		
Test Purpose: This test case verifies the implementation of the rawReadIDs command. Requirements Tested:		
Pre-test conditions:		
<ul style="list-style-type: none"> • Interrogator's antenna is at a fixed and stationary position. • There are no other RFID devices operating in the interrogation area. • Tag0 urn:epc:tag:gid-96:20.300.4000 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>Source.rawReadIDs(null)</code>	The reader should return Tag0 formatted with the current data selector which should contain all supported fields.

Reader Protocol Conformance Requirements Version 1.1

6.11 Requirement command readIDs (TCR11)

TPIId: TCR-11		
Test Purpose: This test case verifies the implementation of the readIDs command. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> • Interrogator's antenna is at a fixed and stationary position. • There are no other RFID devices operating in the interrogation area. • Tag0 urn:epc:tag:gid-96:20.300.4000 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>Source.readIDs(null)</code>	The reader should return Tag0 formatted with the current data selector which should contain all supported fields.

6.12 Requirement command kill (TCR12)

TPIId: TCR-12		
Test Purpose: This test case verifies the implementation of the kill command. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> • Interrogator's antenna is at a fixed and stationary position. • There are no other RFID devices operating in the interrogation area. • Tag0 urn:epc:tag:gid-96:20.300.4000 no access password no kill password. 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>Source.kill(null, null)</code>	The reader should return no error and the tag should be killed and never respond again.

Reader Protocol Conformance Requirements Version 1.1

6.13 Requirement command setReadTimeout (TCR12)

TPIId: TCR-12		
Test Purpose: This test case verifies the implementation of the setReadTimeout command. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> Interrogator's antenna is at a fixed and stationary position. There are no other RFID devices operating in the interrogation area. Tag0 urn:epc:tag:gid-96:20.300.4000 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>Source.setReadCyclesPerTrigger(50000)</code>	If the reader answers with the error "COMMAND_NOT_SUPPORTED" stop the test. This optional command is needed to verify this test.
3	Call command: <code>Source.setReadTimeout(1000)</code>	The reader should return no error. The read timeout is now set to one second. In this time frame 50.000 read cycles can never performed.
4	Call command: <code>Source.readIDs(null)</code>	The command should return after one second, when the read timeout expires. The reader should return then Tag0 once in its report, formatted to the current data selector with all fields supported.

6.14 Requirement command Trigger::getMaxNumberSupported (TCR13)

TPIId: TCR-13		
Test Purpose: This test case verifies the implementation of the static Trigger::getMaxNumberSupported command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>int anz = Trigger::getMaxNumberSupported()</code>	The reader should return at least the amount of 0 to anz when triggers are not supported, else the count of trigger objects which can be created at the reader.

Reader Protocol Conformance Requirements Version 1.1

6.15 Requirement command TagSelector::getMaxNumberSupported (TCR14)

TPId: TCR-14		
Test Purpose: This test case verifies the implementation of the static TagSelector::getMaxNumberSupported command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: int anz = TagSelector::getMaxNumberSupported()	The reader should return at least the amount of 0 to anz when tag selectors are not supported, else the count of tag selector objects which can be created at the reader.

6.16 Requirement command EventType::getSupportedTypes (TCR15)

TPId: TCR-15		
Test Purpose: This test case verifies the implementation of the static EventType::getSupportedTypes command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: EventType events[] = EventType::getSupportedTypes()	The reader should return at least an empty array if events are not supported by this reader, else the list of event types which the reader can generate in its smoothing and event generation stage.

Reader Protocol Conformance Requirements Version 1.1

6.17 Requirement command TriggerType::getSupportedTypes (TCR16)

TPId: TCR-16		
Test Purpose: This test case verifies the implementation of the static TriggerType::getSupportedTypes command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: TriggerType triggerTypes[] = TriggerType::getSupportedTypes()	The reader should return at least an empty array if triggers are not supported by this reader, else the list of trigger types which the reader supports.

6.18 Requirement command FieldName::getSupportedNames (TCR17)

TPId: TCR-17		
Test Purpose: This test case verifies the implementation of the static TriggerType::getSupportedNames command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: FieldName fieldName[] = FieldName::getSupportedNames()	The reader should return a list with the following field names which a reader "SHALL" support: "ReaderEPC", "ReaderName", "ReaderRole", "TagID", "AllSupported". This list can be extended by any further optional field name or a vendor extension.

Reader Protocol Conformance Requirements Version 1.1

6.19 Requirement command Source::getMaxNumberSupported (TCR18)

TPIId: TCR-18		
Test Purpose: This test case verifies the implementation of the static Source::getMaxNumberSupported command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>int iAnz = Source::getMaxNumberSupported()</code>	The reader should return the count of Source objects it can support, or -1 if the number of supported sources is unbound.

6.20 Requirement command NotificationChannel::getMaxNumberSupported (TCR19)

TPIId: TCR-19		
Test Purpose: This test case verifies the implementation of the static NotificationChannel::getMaxNumberSupported command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: <code>int iAnz = NotificationChannel::getMaxNumberSupported()</code>	The reader should return the count of NotificationChannel objects it can support, or -1 if the number of supported notification channels is unbound.

Reader Protocol Conformance Requirements Version 1.1

6.21 Requirement command DataSelector::getMaxNumberSupported (TCR20)

TPId: TCR-20		
Test Purpose: This test case verifies the implementation of the static DataSelector::getMaxNumberSupported command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: int iAnz = DataSelector::getMaxNumberSupported()	The reader should return the count of DataSelector objects it can support, or -1 if the number of supported data selectors is unbound.

6.22 Requirement command TagField::getMaxNumberSupported (TCR21)

TPId: TCR-21		
Test Purpose: This test case verifies the implementation of the static TagField::getMaxNumberSupported command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: int iAnz = TagField::getMaxNumberSupported()	The reader should return the count of TagField objects it can support, or -1 if the number of supported tag fields is unbound.

Reader Protocol Conformance Requirements Version 1.1

6.23 Requirement command ReadPoint.getName (TCR22)

TPId: TCR-22		
Test Purpose: This test case verifies the implementation of the static ReadPoint.getName command. Requirements Tested:		
Pre-test conditions:		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: ReadPoint rp[] = ReaderDevice.getAllReadPoints()	If the reader answers with the error "COMMAND_NOT_SUPPORTED" stop the test. This optional command is needed to verify this test.
3	Call command: String strName = rp[0].getName()	The string strName should contain the name of the first read point of this reader.

6.24 Requirement for a failed command behavior (TCR23)

TPId: TCR-23		
Test Purpose: This test case verifies the behavior of a failed command call. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> Interrogator's antenna is at a fixed and stationary position. There are no other RFID devices operating in the interrogation area. Tag0 urn:epc:tag:gid-96:20.300.4000 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Call command: ReadReport report = Source.readIDs(unknownDataSelector)	The reader should return an error "ERROR_DATASELECTOR_NOT_FOUND". The return value "report" is undefined in this case and depends on the binding which is used. For example: An XML binding would not return a return value for the command, it will return in the binding the error only.

Reader Protocol Conformance Requirements Version 1.1

6.25 Requirement for a unsupported handshake (TCR24)

TPIId: TCR-24		
Test Purpose: This test case verifies the behavior of a handshake where to host request an unsupported property of the reader. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> • Interrogator's antenna is at a fixed and stationary position. • There are no other RFID devices operating in the interrogation area. 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Send the Handshake with RP Version 2.2: RPS122X1X1NA0000END1	The reader should reply with the answer: RPR1NO11X1X1OK00END1 Further the reader should close the connection.

6.26 Requirement for an invalid handshake format (TCR25)

TPIId: TCR-25		
Test Purpose: This test case verifies the behavior of a undefined string send instead of a valid handshake string. Requirements Tested:		
Pre-test conditions: <ul style="list-style-type: none"> • Interrogator's antenna is at a fixed and stationary position. • There are no other RFID devices operating in the interrogation area. 		
Step	Step description	Expected results
1	Power on the reader.	Wait till reader boot up finished.
2	Send a undefined string instead of a valid handshake: XYZABC1234	The reader should close the connection without any reply.