Interoperability Test System for EPC Compliant Class-1 Generation-2 UHF RFID Devices

Interoperability Test Methodology

Release 2.0.1, Ratified, Feb 2016

Document Summary

Document Item	Current Value
Document Name	Interoperability Test System for EPC Compliant Class-1 Generation-2 UHF RFID Devices
Document Date	Feb 2016
Document Version	2.0
Document Issue	1
Document Status	Ratified
Document Description	Interoperability Test Methodology

Revision History

Release	Date of Change	Changed By	Summary of Change
2.0.0	2015-02-16	Ted Osinski	First draft for G2V2 IOP Test Spec
2.0.0	2015-07-03	Michael Koch	Rearrange for readability
2.0.0	2015-07-07	Ted Osinski/M	Minor edits
2.0.1	2016-02-11	Ted Osinski	Added "command" parameter to all directives to enable Application IOP and Extended IOP testing Created new directives for Application IOP and optional commands and removed "command" parameter from Core IOP directives. This will make Core IOP directives identical to G2V1.

Copyright Notice

Disclaimer

Table of Contents

Figures

Tables

1 Scope

This document specifies the design of an Interoperability test system for testing that end-to-end functionality between two communicating RFID hardware devices as required by "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHZ – 960 MHz Version 2.0.0" (hereinafter to be referred as the Protocol).

The RFID devices under test are interrogators, also known as readers, printers, also known as encoders, and one or more tags, also known as labels. Tags can be passive, meaning that they receive all of their operating energy from the interrogator's RF waveform. They can also be semi-passive, or active provided the utilized integrated circuit (IC) is compliant to the above referenced specification. The protocol is interrogator-talks-first (ITF), meaning that a tag modulates its antenna reflection coefficient with an information signal only after being directed to do so by an interrogator.

Interoperability certification testing is conducted after conformance certification testing and is intended to complement conformance testing.

The theme of interoperability testing is "a walk through Gen2 state diagram". This approach will minimize the number of test cases needed. It is important to note that test cases for each Device Under Test (DUT) will be selected from the pool of available test cases based on DUT's implemented features and features implemented on the QE. Therefore, test cases for a particular DUT and data used for each test will be different for each DUT.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] EPCglobal, Inc. "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHZ – 960 MHz. Version 2.0.0

[2] EPCglobal, Inc. "EPC™ Tag Data Standards"

[3] EPCglobal™ (2004): FMCG RFID Physical Requirements Document (draft)

[4] ISO/IEC 15961: Information technology-Radio frequency identification (RFID) for item management-Data protocol: application interface.

[5] ISO/IEC 15962: Information technology-Radio frequency identification (RFID) for item management-Data protocol: data encoding rules and logical memory functions.

[6] ISO/IEC 15963: Automatic Identification- Radio Frequency Identification for item management-Unique identification for RF tag.

[7] ISO/IEC 9646-1: Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts.

[8] ISO/IEC 9646-2: Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification.

[9] ETSI ETS 300 406: Methods for Testing and Specification (MTS); Protocol and Profile Conformance Testing specifications; Standardization methodology.

[10] ETSI TS 102 237-1 v 4.1.1 (2003-12): Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Interoperability test methods and approaches; Part 1: Generic approach to interoperability testing.

[11] Interoperability Test System for EPC compliant devices Class-1 Generation-2 UHF RFID devices. Requirements for the Interop tester v1.0 CETECOM

3 Definitions and Abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Interoperability: Ability of two or more systems or devices to exchange information using the same communication protocol.

Interoperability testing: Activity of testing end-to-end functionality between (at least) two communicating systems as required by the standard(s) on which those systems are based.

Reference specification: A standard, which specifies a base specification, or a set of base specifications, or a profile, or a set of profiles, and for conformance against which test specifications are written.

Interrogator samples for interoperability testing: A device which combines both transmission and reception capabilities within a single housing. Its components are an antenna, a RFID reader and suitable control software to evaluate interrogator performance.

Tag samples for interoperability testing: RFID tags contain an antenna and an electronic microchip to enable them to receive and respond to radio-frequency queries from an RFID transceiver. A minimum sample size is required to complete testing since some will be permanently altered in the course of testing.

Test Purpose (TP): Easy-to-read description of each test, concentrating on the meaning of the test rather than detailing how it may be achieved. The Test Purpose is derived from the reference specification and focuses on testing a specific functionality of the DUT (Equipment Under Test i.e. reader or tag) that can be affected at the user interfaces offered by the SUT (System Under Test).

Test Suite: A major subset of the Gen2 protocol. A test suite is verified by running a number of like test cases. A simple scripting language can be used to sequence through the test cases. The script links to a reader application that issues commands and collects responses from the tag. Success/failure for each test case is determined by comparing the responses to the expected responses.

Test Case: A fundamental functionality within the Gen2 protocol, for instance, reading the Access password. Test cases are grouped in order to verify a test suite. All test cases within a test suite must be successful in order for the suite to be declared successfully verified. Test cases may require more than one reader command to be verified. For example, a Write test case is verified only after a subsequent Read. The Write and Read can be mated in a script to accomplish this.

Implementation Statement (IS): A checklist of the capabilities/functionalities supported by the DUT is used to select and parameterize test cases and as an indicator of interoperability between different products.

Implementation eXtra Information for Testing (IXIT): Contains additional information (e.g., specific addresses, timer values, etc.) necessary for testing.

Device Under Test (DUT): An interrogator, a tag or a tag population. The subject of the test may be a single DUT, which is testing against a QE.

Qualified Equipment (QE): A device that has been shown, by rigorous and well-defined testing, to operate with other equipment and adhere to the protocol.

System Under Test (SUT): One or more DUTs and /or QEs. For the purposes of the present document, the SUT may comprise one or more QEs and a single DUT, or possibly two DUTs, depending on the selected test scenario. In all cases, the test scenario shall be comprised of at least one interrogator and one tag.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

A	Access
AFI	Application Family Identifier
AP	Access password
ATS	Application IOP Test Script
AU	Authenticate
BLF	Backscatter Link Frequency
BP	BlockPermalock
BW	BlockWrite
C	Challenge or Complete (change complete memory), depending on context
CR	Crypto
CRC	Cyclic Redundancy Check
DUT	Device Under Test
E	EPC Memory
EN-RT	European Normative Requirements Table
EPC	Electronic Product Code
ETS	Extended IOP Test Script
ETSI	European Telecommunications Standards Institute
FO	File Open
Handle	16-bit Tag-authentication number
I	Inventory
IOP	Interoperability
IS	Implementation Statement
ITF	Interrogator Talks First (Reader Talks First)
IXIT	Implementation eXtra Information for Testing
K	Kill
KP	Kill password
KU	Key Update
L	Locked state or Lock functionality, depending on context
M	Mandatory, shall be implemented under all circumstances
MH	Multiple Homogeneous populations of tags
MM	Multiple Mixed population of tags
MTR	Message Transfer
N	Long, change longer than memory bank
NAP	No Access Password
NSI	Numbering System Identifier
NZ	Write non-zero value
O	Optional, may be provided, but if provided shall be implemented in accordance with the requirements

O.n	This status is used for mutually exclusive or selectable options among a set. The integer "n" shall refer to a unique group of options within the EN-RT. A footnote to the EN-RT shall explicitly state what the requirement is for each numbered group. For example, "It is mandatory to support at least one of these options", or, "It is mandatory to support exactly one of these options".
P	Partial, change portion of memory
PC	Protocol Control
PL	Permanently locked (Permalocked) or Permalock functionality, depending on context
PU	Permanently unlocked (Permaunlocked) or Permaunlock functionality, depending on context
QE	Qualified Equipment
R	Read
RFID	Radio-Frequency IDentification
RFU	Reserved for Future Use
RN16	16-bit Random or pseudo-Random Number
RNG	Random or pseudo-Random Number Generator
SC	SecureComm
SI	Select/Inventory
SQ	Select/Query
SUT	System Under Test
T	TID memory
TBD	To Be Defined
TC	Test Case
TID	Tag-IDentification or Tag-Identifier, depending on context
TP	Test Purpose
TS	Test Suite or Test Script for Core IOP Test Category, depending on context
TSS	Test Suite Structure
U	User memory, Unlock functionality or Unlocked state, depending on contextUHF Ultra High Frequency
UN	Untraceable
V	Valid
W	Write
Word	16 bits
Z	Write zero value

4 Interoperability Test Structure

4.1 Overview

Interoperability testing can be defined as the functional testing of a product against another operational product according to a set of test specifications. Therefore, testing of individual commands is not the goal of the interoperability.

Interoperability tests are based on functionality as experienced by the user (i.e., they are not necessarily specified at the protocol level). Also the interoperability tests are performed at interfaces that offer normal user control and observation.

The described test system will provide the user's interfaces and other facilities for interoperability testing and reduce testing time. As a means of improving testing coverage, efficiency and consistency, a scripting language is specified that allows test cases to be concatenated and run automatically. An output file is created that contains the test results thereby easing documentation.

The interoperability test system will use Qualified Equipment (QE). Qualified Equipment may be a reader, a tag or a population of tags depending on the device under test. The Device Under Test will be tested against a QE (DUT against QE).

Interoperability testing will not test for backward compatibility.

4.2 Implementation Statement (IS/IXIT)

The IS is generated by a manufacturer according to a standardized template that details all supported mandatory and optional features for the DUT.

The IS is a checklist of the capabilities/functionalities supported by the DUT is used to select and parameterize test categories and test cases and as an indicator of interoperability between different products.

The IXIT is created by a manufacturer using a standardized template that contains additional information (e.g., specific addresses, timer values, application commands, parameters etc.) necessary for interoperability testing.

4.3 Interoperability (IOP) Test Categories

Based on the submitted Implementation Statement, a number of test cases will be generated in applicable test categories. Each category will have test cases within various test suites.

The test cases may be subject to revisions during the validation process.

Three major test categories are defined as follows:

- **Core IOP** - Mandatory for all devices
- **Application IOP** - Mandatory for devices designed to meet Annex N Requirements of the Protocol
- **Extended IOP -** Optional for devices that request to be tested for other features such as core access optional commands or file system commands

4.3.1 Core IOP Test Category

The tests in this category will test the transition of a DUT from one state to another. This will be accomplished by the reader issuing core mandatory commands. The core mandatory commands (*Reg_RN, Read, Write, Lock and Kill*) will be used for all tests. It's important to note that the test scripts will not be invoking individual commands directly.

4.3.2 Application IOP Test Category

A tag or a reader will be tested in context of implemented functionality as defined in Annex N of the Protocol. It's important to note that the test scripts will not be invoking individual commands directly.

Table 4-1 Application Categories

Application (per Annex N)	Optional Commands implemented
Alteration EAS	*Untraceable, Access*
Tag Alteration (Core)	*Untraceable, Access*
Tag Alteration (Challenge)	*Untraceable, Access, Challenge*
Tag Alteration (Authenticate)	*Untraceable, Access, Authenticate*
Tag Alteration (Full)	*Untraceable, Access, Challenge, Authenticate, BlockWrite, SecureComm*
Consumer Electronics	*BlockPermalock, Acccess, FileOpen*

4.3.3 Extended IOP Test Category

This test category will include all optional commands that can be tested and are requested to be tested by the submitter). Testing for this category will involve specifying a command to be tested in the script file. A command will be passed to interoperability program and reader will be instructed to execute it.

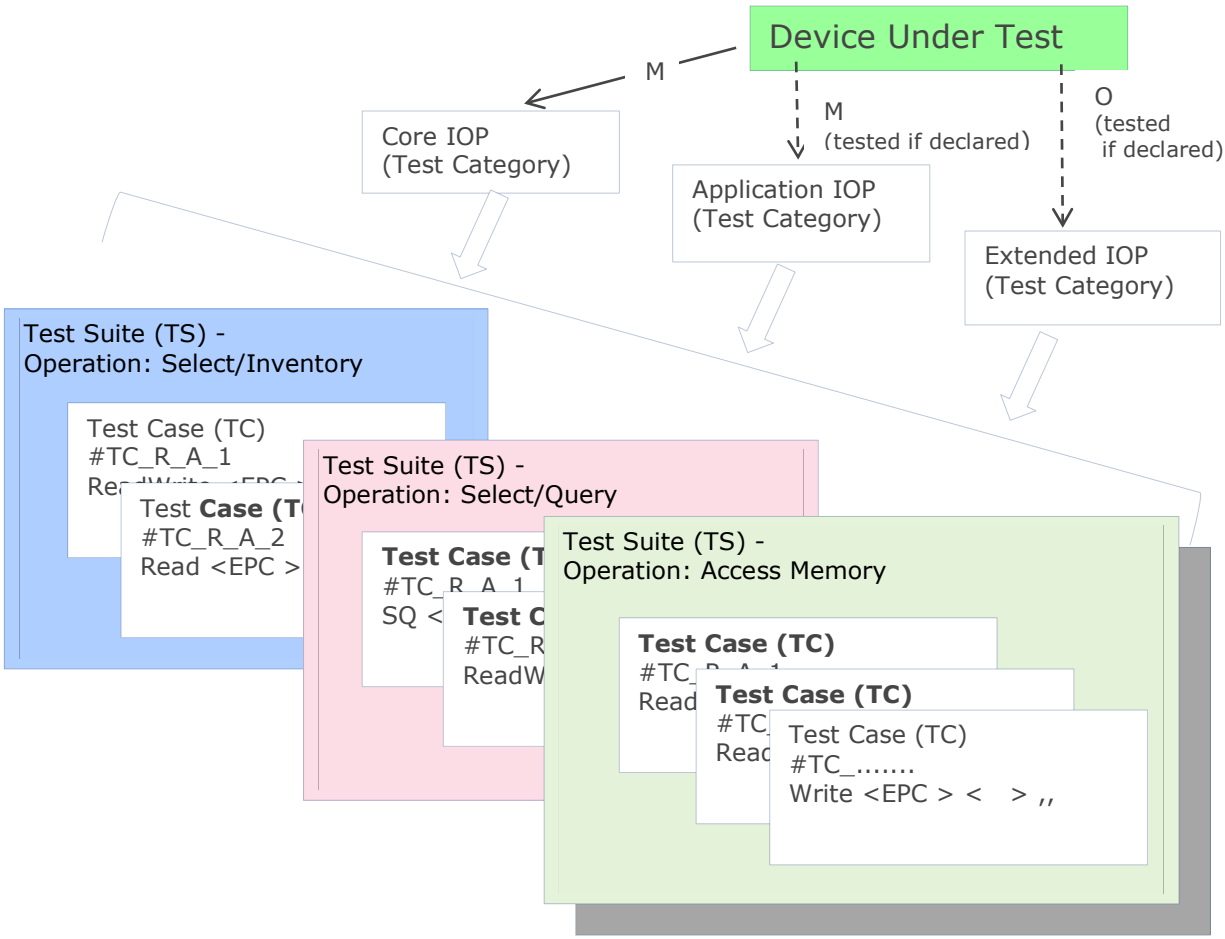
- Optional Core commands
  (*Access, BlockWrite, BlockErase, BlockPermalock, Untraceable*)
- Optional File commands
  (*FileOpen, FileList, FileSetup, FilePrivilege,)*
- Optional Security (Crypto) commands
  (Authenticate, SecureComm, KeyUpdate, TagPrivilege)

4.4 IOP Test Case Suites and Test Cases

The fundamental unit of interoperability testing is a Test Case. It consists of a directive and parameters that are used to pass data to a reader. Test Cases are grouped into functionally cohesive Test Suites. TCs can be repeated in Test Suites.

Test Categories are declared by a vendor. The tailored Test Suites are executed for each Test Category.

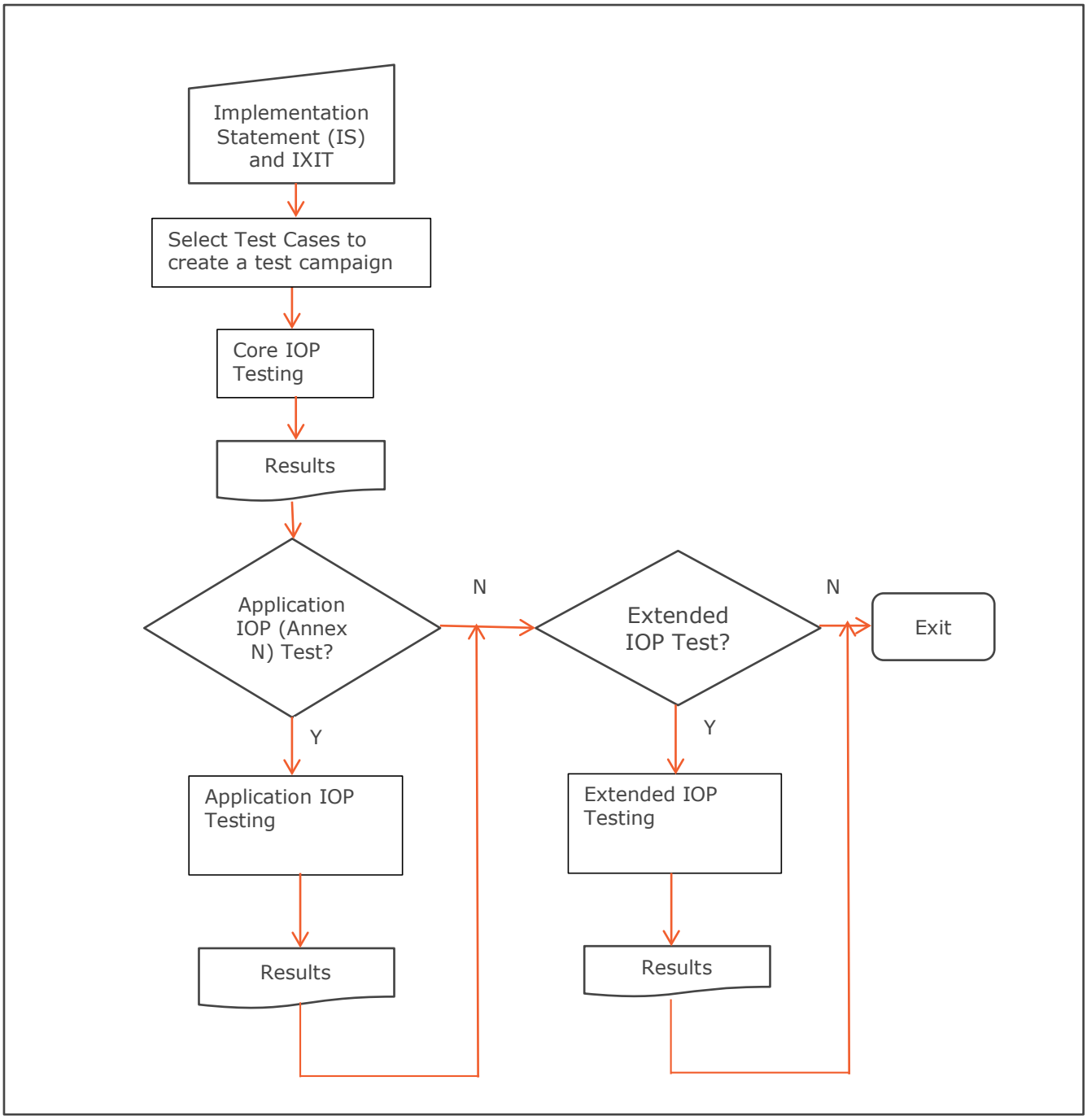
Figure 4-1 Test Categories and Test Suites



5 Executing IOP Tests

The following flowchart depicts selection of test cases from the Implementation Statement (IS) declarations and the execution of the test cases.

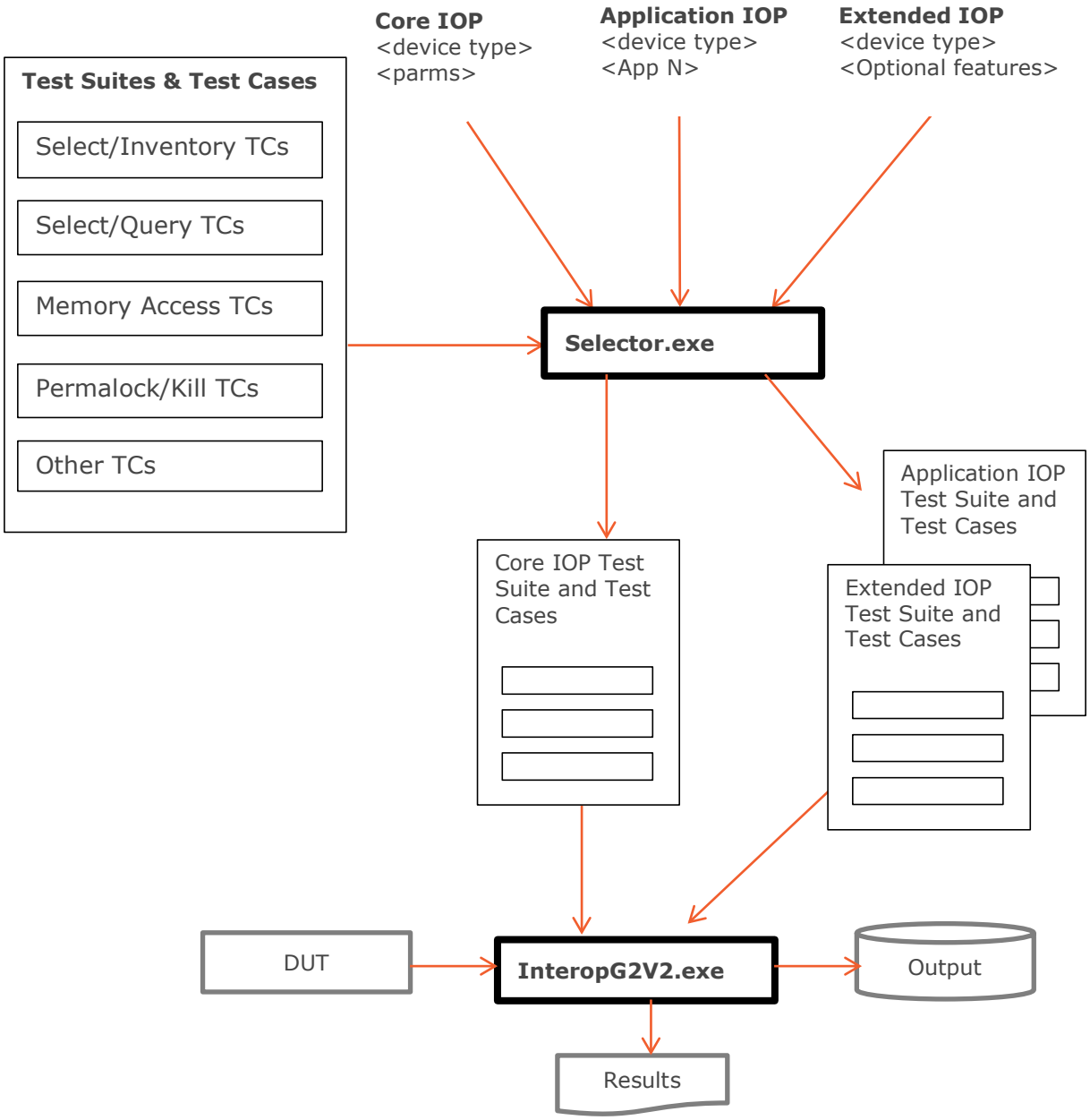
Figure 5-1 Executing IOP Tests



5.1 Automating Selection of Test Cases

Due to a large number of test cases, parameters and the three interoperability categories necessitates an automatic procedures to select and assemble test cases for a submitted DUT (Device Under Test). A computer program should perform the selection of test cases to improve accuracy and reduce time.

Figure 5-2 Automating Selection of Test Cases



5.2 IOP Operations

Test Suites are created by assembling a group of test cases that will conduct operations on a tag or on a reader.

For interoperability coverage, the following Operations/Test Suites have been defined; Select/Inventory, Select/Query, Memory Access, Permalock/Kill, WriteRead and Special. They are loosely based on the Operations as defined in section 6.3.2.8 Managing Tag Population of the Protocol. The set of Test Suites is run for a given reader/tag air interface condition. The set of parameters that defines the air interface settings is called the Mode. A Mode defines the reader-to-tag characteristics (Mod (short for modulation) type, Tari, PIE, mask type) and the tag-to-reader characteristics (BLF, M, DR, TRext). If a reader supports more than one mode, the bounding modes (longest and shortest Tari's) shall be tested. The pseudo-code below illustrates the testing hierarchy used to show interoperability.

Mode (Mod type/Tari/PIE/mask type/BLF/M/DR/TRext)

Select/Inventory (Non-select inventory, Select inventory)

Multi-tag (homogeneous, mixed)

Memory (EPC, TID, User/File)

Single tag

Memory (EPC, TID, User/File)

Lock (Unlocked, Locked, Permaunlocked, Permalocked)

END Select/Inventory

Select/Query (memory bank, session flags, actions, mask, truncate)

END Select/Query

Memory Access (unsecured, secured)

Memory (Passwords, EPC, TID, User/File)

Lock (Unlocked, Locked, Permaunlocked, Permalocked)

END Memory Access

Permalock/Kill

Memory (Passwords, EPC, TID, User/File)

Lock (Permaunlocked, Permalocked)

END Permalock/Kill

Special

Slot counter (ACK, don't ACK)

END Special

END Mode

The Select/Inventory suite verifies that a sub-population of tags can be selectively inventoried and that the Select command elicits the proper response from the tag. Multi and single tag testing is performed to verify the Select and Inventory functions within the protocol.

The Select/Query suite verifies that a sub-population of tags can be selectively inventoried and queried, and that the correct response is received from the tag. Multi and single tag testing is performed to verify the Select and Query functions within the protocol.

The Memory Access suite confirms that tag memory can be appropriately accessed. Memory access through the secured and unsecured state diagram paths are exercised. Access is evaluated with memories pre-configured in the each of their possible four states; unlocked, locked, permaunlocked or permalocked.

The Permalock/Kill suite tests the memories in their permaunlocked and permalocked states. Once permalock testing is completed, the tag's Kill functionality is verified. This suite is separated so that the number of tags permanently altered is minimized.

The Special suite is reserved for miscellaneous tests that don't fall in the other suites and are specific to V2.0.0 of the specification.

5.3 IOP Test Scripts

5.3.1 Scripts Overview

The test cases (TCs) are coded in Perl scripts. Perl is a computer neutral language. Reader vendors may code interoperability program in any programming language.

5.3.2 Script Language Syntax

The Reader vendor provides a PC application that runs a Script File and produces an Output File. This application is invoked using the following syntax, where "InteropG2V2" is the call to the application and input file is the Script filename and output file is the Output filename.

InteropG2V2 <input file> <output file>

The input file (Script File) is a text file comprised of directives from a scripting language described below. The Script File shall be read by the application and each line executed in order. Results shall be written both to the screen and to the output file (Output File) for documentation and as inputs to the Software Manager for parsing and report generation. Examples of a Script and Output File are provided in this specification.

5.3.3 Reader Directives

The script has directives in a specific format. The directives instruct the IOP program on functionality that is being tested and the IOP program communicates them to a reader.

The example of the scripts is listed below.

```
# This section initializes the tag. This is a comment
#The line below contains directive
Write pass 00000000 apass     # This is a comment
Write pass 00000000 kpass
#
# End of initialize tag
Read fail 00000000 apass EEEEEEEE
# AP = 00000000, U; AP read w/ incorrect AP; TC = R_AP_2
# This TC reads the correct AP when the AP is zero, using an incorrect AP.
```


5.4 Interoperability Test Environment

5.4.1 General Characteristics

This section introduces general characteristics of test environment used to test the test cases defined in this document. The characteristics here described apply to all test cases detailed in the following sections:

- The test site shall be on a reasonably flat surface or ground.
- For inventory and access operations, the distance between the reader and tag shall be as specified by the tag manufacturer for the particular RFID application under evaluation. Reader transmit power shall be sufficient to provide the specified power to the tag over that range. In the case of a mixed population of tags, the distance will be set to the minimum tag distance requirement. Note that the distance requirement may be different for inventory, read, write, lock, and kill operations. To facilitate continuous testing via script, the minimum distance across all operations may be used for all tests.

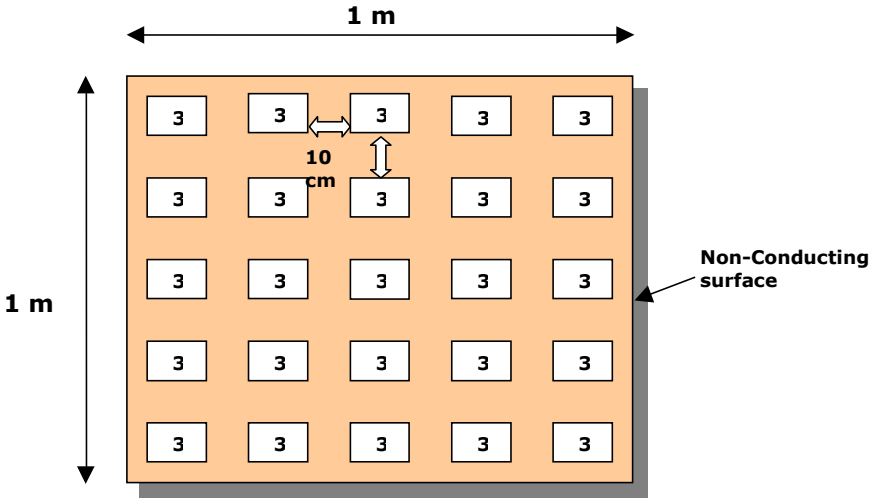
The next items describe general statements concerning the interrogator placement in all test sites:

- Interrogator shall be at fixed and stationary position.
- The interrogator antenna shall be placed vertical on a non-conducting support.
- The height of the interrogator antenna shall ensure the correct operation of the entire interrogator.
- If the interrogator can use two or more antennas simultaneously, just one antenna shall be operating and connected to the interrogator.
- For ensuring an interference free environment, no other devices or interrogators shall be operating, at the same frequency range as the SUT, inside the test site area.

5.4.2 Multiple-tag Setup

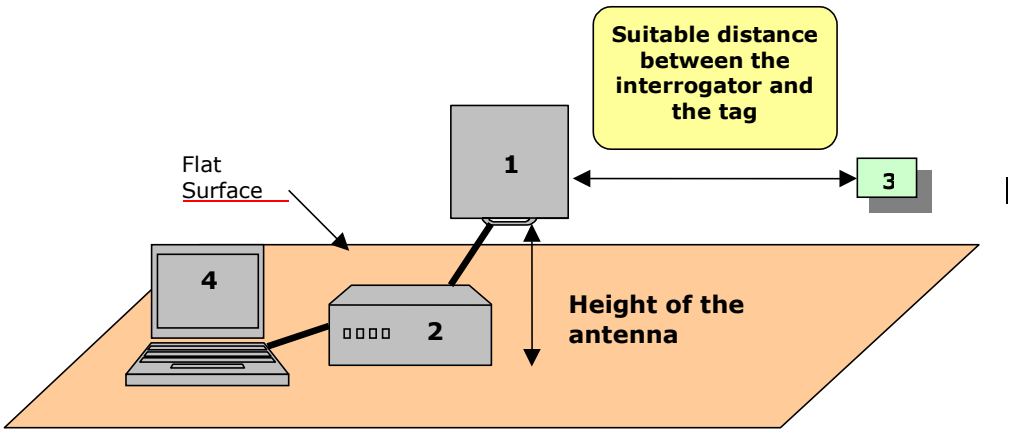
The multiple-tag setup may be a flat surface of a non-conducting material with maximum area of 1 m x 1 m. The multiple-tag setup contains a group of tags from the same (homogeneous) or different (mixed) label or inlay manufacturers. This tags shall contain a class 1 Gen 2 IC. The tags shall be evenly distributed and maintain a separation between them of not less than 10 cm. The maximum number of tags in the setup shall be 25. Figure 5-3 depicts the multiple-tag setup.

Figure 5-3 Multi-tag Board



5.4.3 Test Setup for Single Tag Selection

Figure 5-4 Test Setup for Single Tag Selection



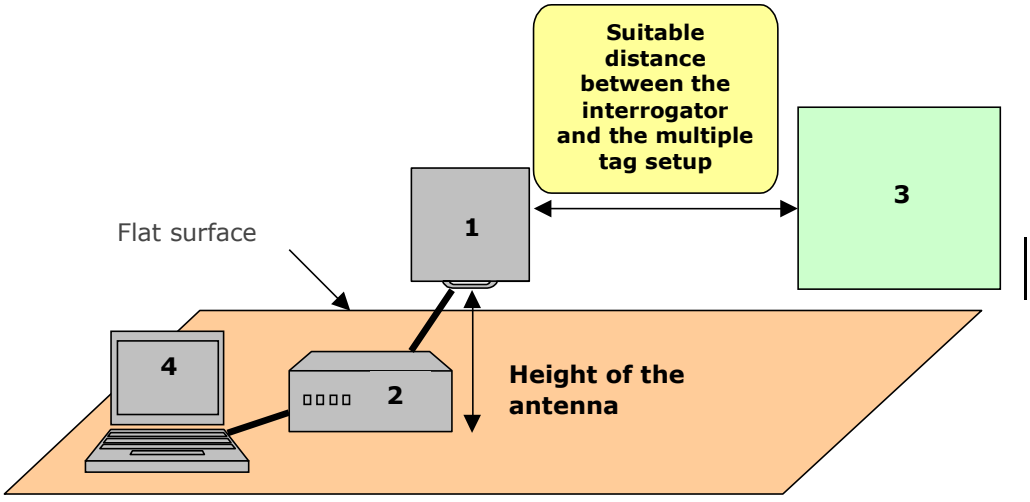
1 Interrogator antenna
2 Interrogator
3 Tag
4 Host

Description of the tag placement:

- The tag shall be at fixed and stationary position.
- The geometrical centre of the tag shall be aligned horizontally with the geometrical centre of the interrogator antenna and the tag orientation facing the antenna.
- Tag placement shall be on a non-conducting material, for example cardboard. The tag should be placed facing the interrogator antenna.

5.4.4 Test Setups for Multiple Tag Selection

Figure 5-5 Test Setup for Multiple Tag Selection



1 Interrogator antenna
2 Interrogator
3 Multiple-tag setup
4 Host

Description:

- The multiple-tag setup shall be at fixed and stationary position.
- The geometrical centre of the multiple-tag setup shall be aligned horizontally with the geometrical centre of the interrogator antenna and its orientation facing the antenna.
- Test Site for homogeneous multi-tag testing shall consist of a population of tags from the same vendor. This is referred to as a homogeneous multi-tag population.
- Test Site for mixed multi-tag testing shall consist of a population of tags from the different vendors. This is referred to as a mixed multi-tag population.

5.5 File Folders for Scripts

Scripts are grouped and stored in file folders according to test category. The folders are defined as follows:

Scripts in the folders will be modified upon submission of Implementation Statement (IS) and Implementation eXtra Information for Testing (IXIT).

5.5.1 Folder Definitions for Core IOP Test Category.

TS= Standard test scripts for tags and interrogators with an access password, permalocked TID and no User-memory.

TS NAP= Test scripts with permalocked TID and No Access Password

> **NOTE:** some scripts do not have to be performed on these tags because they only have to do with non-zero Access Passwords.

TS U_TID= Tag/Interrogator test scripts where the tags have unlockable TID.

TS_U= Tag and Interrogator User scripts where the tags have User Memory or File Systems

5.5.2 Folder Definitions for Application IOP Test Category.

ATS= Standard test scripts for tags and interrogators with an access password, permalocked TID and no User-memory.

ATS NAP= Test scripts with permalocked TID and No Access Password

> **NOTE:** some scripts do not have to be performed on these tags because they only have to do with non-zero Access Passwords.

ATS U_TID= Tag/Interrogator test scripts where the tags have unlockable TID.

ATS_U= Tag and Interrogator User scripts where the tags have User Memory or File Systems

5.5.3 Folder Definitions for Extended IOP Test Category.

ETS= Standard test scripts for tags and interrogators with an access password, permalocked TID and no User-memory.

ETS NAP= Test scripts with permalocked TID and No Access Password

> **NOTE:** some scripts do not have to be performed on these tags because they only have to do with non-zero Access Passwords.

ETS U_TID= Tag/Interrogator test scripts where the tags have unlockable TID.

ETS_U= Tag and Interrogator User scripts where the tags have User Memory or File Systems

6 Test Cases (Normative)

6.1 Naming Conventions

Read, Write, Lock, Select, and Inventory test cases are listed in the tables with this section. A test case identifier is assigned to each test case using the approximate syntax described in the following table. All of the test cases shall be exercised within at least one of the Test Suites for each mode to assure interoperability coverage.

Table 6-1 Approximate Test Case Identifier naming convention scheme

Identifier: **<functionality under test>_<memory targeted>_<memory state>_<memory action>_<nn>**	
<functionality under test>	I (Inventory) SI (Select then Inventory) SQ (Select then Query) W (Write) L (Lock) U (Unlock) PL (Permalock) PU (Permaunlock) K (Kill) R (Read) CR (Crypto) UN (Untraceable) FO (File Open) C (Challenge) A (Access) AU (Authenticate) BW (BlockWrite) BP (BlockPermalock) KU (Key Update) SC (SecureComm)
<memory targeted>	AP (Access Password) KP (Kill Password) E (EPC Memory) T (TID Memory) U (User Memory) File_0 File>0 CRC (CRC-16) MH (Multiple homogenous population) MM (Multiple mixed population)
<memory state>	U (Unlocked) L (Locked) PL (Permalocked) PU (Permaunlocked) P (Partial, change portion of memory) C (Complete, change complete memory) N (Long, change longer than memory bank)
<memory action>	Z (Write zero value) NZ (Write non-zero value)
<nn>	sequential number (1-99)

6.2 Test Cases List

The following tables list the available test cases. They are grouped and placed into script files (*.txt) that become inputs to an "InteropG2V2.exe" program. The selection of test cases is dependent on the features supported by a DUT and a QE reader/tag. The naming conventions for (*.txt) scripts files such as Access_Memory.txt are informational and can be changed.

Table 6-2 Inventory Single & Multiple Test Cases

Inventory_single.txt	Inventory_multiple.txt
Test Cases	Test Cases
I	I_MH
I_L	SI_MH_E
SI_E	SI_MH_T
SI_E_L	SI_MH_U
SI_T	I_MM
SI_T_L	SI_MM_E
SI_U	SI_MM_T
SI_U_L	SI_MM_U

Table 6-3 User Memory Test Cases

User memory		Verify user memory operations are correctly performed
Test cases	Lock	Action
R_U_C		User complete
R_U_P		User partial
R_U_P_L	L	User partial
R_U_P_PU	PU	User partial
R_U_P_PL_1	PL	User partial; Access password zero
R_U_P_PL_2	PL	User partial; Access password non-zero
W_U_C		User complete
W_U_P		User partial
W_U_L_P	L	User partial
W_U_PU_P	PU	User partial
W_U_PL_P_1	PL	User partial; Access password zero
W_U_PL_P_2	PL	User partial; Access password non-zero
L_U		User memory
PU_U		User memory
U_U_L	L	User memory
PL_U_L_1	L	User memory; Access password zero
PL_U_L_2	L	User memory; Access password non-zero
U_U_PU	PU	User memory
L_U_PU	PU	User memory
PU_U_PU	PU	User memory
U_U_PL_1	PL	User memory; Access password zero
U_U_PL_2	PL	User memory; Access password non-zero
L_U_PL_1	PL	User memory; Access password zero
L_U_PL_2	PL	User memory; Access password non-zero
PU_U_PL_1	PL	User memory; Access password zero
PU_U_PL_2	PL	User memory; Access password non-zero

Table 6-4 Access Test Cases

Access		Verify access operations are correctly performed		
Test cases	Lock	AP Action	AP State	AP
R_AP_1		-	zero	Correct
R_AP_2		-	zero	Incorrect
R_AP_3		-	non-zero	Correct
R_AP_4		-	non-zero	Incorrect
R_AP_5		-	non-zero	None
R_AP_L_1	L	-	zero	Correct
R_AP_L_2	L	-	zero	Incorrect
R_AP_L_3	L	-	non-zero	Correct
R_AP_L_4	L	-	non-zero	Incorrect
R_AP_L_5	L	-	non-zero	None
R_AP_PU_1	PU	-	zero	Correct
R_AP_PU_2	PU	-	zero	Incorrect
R_AP_PU_3	PU	-	non-zero	Correct
R_AP_PU_4	PU	-	non-zero	Incorrect
R_AP_PU_5	PU	-	non-zero	None
R_AP_PL_1	PL	-	non-zero	Correct
R_AP_PL_2	PL	-	non-zero	None
W_AP_Z		Zero	non-zero	Correct
W_AP_NZ_1		non-zero	zero	Correct
W_AP_NZ_2		non-zero	non-zero	Correct
W_AP_NZ_3		non-zero	non-zero	Incorrect
W_AP_NZ_4		non-zero	non-zero	None
W_AP_L_NZ_1	L	non-zero	zero	Correct
W_AP_L_NZ_2	L	non-zero	zero	Incorrect
W_AP_L_NZ_3	L	non-zero	non-zero	Correct
W_AP_L_NZ_4	L	non-zero	non-zero	Incorrect
W_AP_L_NZ_5	L	non-zero	non-zero	None
W_AP_PU_Z	PU	Zero	non-zero	Correct
W_AP_PU_NZ_1	PU	non-zero	zero	Correct
W_AP_PU_NZ_2	PU	non-zero	non-zero	Correct
W_AP_PU_NZ_3	PU	non-zero	non-zero	Incorrect
W_AP_PU_NZ_4	PU	non-zero	non-zero	None
W_AP_PL_NZ_1	PL	non-zero	non-zero	Correct
W_AP_PL_NZ_2	PL	non-zero	non-zero	None
L_AP_1		-	zero	Correct
L_AP_2		-	zero	Incorrect
L_AP_3		-	non-zero	Correct

Access		Verify access operations are correctly performed		
L_AP_4		-	non-zero	Incorrect
L_AP_5		-	non-zero	None
PU_AP		-	zero	Correct
U_AP_L_1	L	-	zero	Correct
U_AP_L_2	L	-	zero	Incorrect
U_AP_L_3	L	-	non-zero	Correct
U_AP_L_4	L	-	non-zero	Incorrect
PL_AP_L_1	L	-	non-zero	Incorrect
PL_AP_L_2	L	-	non-zero	None
PL_AP_L_3	L	-	zero	Correct
L_AP_PU_1	PU	-	zero	None
L_AP_PU_2	PU	-	non-zero	Correct
L_AP_PU_3	PU	-	non-zero	None
PU_AP_PU_1	PU	-	zero	None
PU_AP_PU_2	PU	-	non-zero	Correct
PU_AP_PU_3	PU	-	non-zero	None
U_AP_PL_1	PL	-	zero	Correct
U_AP_PL_2	PL	-	non-zero	Correct
U_AP_PL_3	PL	-	non-zero	None
L_AP_PL_1	PL	-	zero	Correct
L_AP_PL_2	PL	-	non-zero	Correct
L_AP_PL_3	PL	-	non-zero	None
PU_AP_PL_1	PL	-	zero	Correct
PU_AP_PL_2	PL	-	non-zero	Correct
PU_AP_PL_3	PL	-	non-zero	None

Table 6-5 Access Memory – File System

Additional test cases for testing File System.

Access_memory_FS.txt	Write_read_FS.txt
TCs – File System	**TCs –File System**
R_AP_6	R_KP_1
R_AP_L_6	R_KP_1
W_AP_NZ_5	W_KP_Z_1
W_AP_L_NZ_6	W_KP_L_NZ_1
L_AP_6	L_KP_1
U_AP_L_5	U_KP_L_1
	K_INZ_L_1
	R_E_L_P_1
	W_E_L_P_1
	L_E_1_1
	U_E_L_1
	R_T_C_1
	R_T_P_1

Access_memory_FS.txt	Write_read_FS.txt
	W_T_C_1
	L_T_1
	U_T_L_1
	R_U_C_1
	R_U_P_1
	W_U_C_1
	L_U_1
	U_U_L_1

Table 6-6 Select/Inventory Test Cases

Select/Inventory		
Test cases	Lock	Action
Multi-tag		Verify ability to inventory all tags or a selected sub-population
I_MH		Non-select inventory homogeneous
SI_MH_E		Select EPC complete homogeneous
SI_MH_T		Select TID complete homogeneous
SI_MH_U		Select User complete homogeneous
I_MM		Non-select inventory mixed
SI_MM_E		Select EPC complete mixed
SI_MM_T		Select TID complete mixed
SI_MM_U		Select User complete mixed
Single tag		Verify ability to Select and Inventory with memories in various lock states
I		Non-select inventory
I_L	L	Non-select inventory
I_PU	PU	Non-select inventory
I_PL	PL	Non-select inventory
SI_E		Select EPC complete
SI_E_L	L	Select EPC partial
SI_E_PU	PU	Select EPC partial
SI_E_PL	PL	Select EPC partial
SI_T		Select TID complete
SI_T_L	L	Select TID partial
SI_T_PU	PU	Select TID partial
SI_T_PL	PL	Select TID partial
SI_U		Select User complete
SI_U_L	L	Select User partial
SI_U_PU	PU	Select User partial
SI_U_PL	PL	Select User partial

Table 6-7 Permalocked Test Cases

Permalocked_APZ.txt	Permalocked_APNZ.txt	PermaUnlocked.txt	PermaLocked_L.txt
Test Cases	Test Cases	Test Cases	Test Cases
R_KP_PL_1	R_KP_PL_2	R_KP_PU	PL_AP_L_1
W_KP_PL_NZ_1	W_KP_PL_NZ_2	W_KP_PU_NZ	PL_AP_L_2
PL_KP	PL_KP_L_2	PU_KP	
PL_KP_L_1	U_KP_PL_2	U_KP_PU	
U_KP_PL_1	L_KP_PL_2	L_KP_PU	
L_KP_PL_1	PU_KP_PL_2	PU_KP_PU	
PU_KP_PL_1	R_E_PL_P_2	K_INZ_PU	
K_INZ_PL	W_E_PL_P_2	R_E_PU_P	
K_NZ	PL_E_L_2	W_E_PU_P	
R_E_PL_P_1	U_E_PL_2	PU_E	
W_E_PL_P_1	L_E_PL_2	U_E_PU	
PL_E_L_1	PU_E_PL_2	L_E_PU	
U_E_PL_1	R_T_PL_P_2	PL_E_PU	
L_E_PL_1	W_T_PL_P_2	PU_E_PU	
PU_E_PL_1	PL_T_L_2	R_T_PU_P	
R_T_PL_P_1	U_T_PL_2	W_T_PU_P	
W_T_PL_P_1	L_T_PL_2	U_T_PU	
PL_T_L_1	PU_T_PL_2	L_T_PU	
U_T_PL_1	R_U_P_PL_2	PL_T_PU	
L_T_PL_1	W_U_PL_P_2	PU_T_PU	
PU_T_PL_1	PL_U_L_2	R_U_P_PU	
R_U_P_PL_1	U_U_PL_2	W_U_PU_P	
W_U_PL_P_1	L_U_PL_2	PU_U	
PL_U_L_1	PU_U_PL_2	U_U_PU	
U_U_PL_1	R_AP_PL_1	L_U_PU	
L_U_PL_1	R_AP_PL_2	PL_U_PU	
PU_U_PL_1	W_AP_PL_NZ_1	R_AP_PU_1	
PL_AP_L	W_AP_PL_NZ_2	R_AP_PU_2	
U_AP_PL_1	L_AP_PU_2	R_AP_PU_3	
L_AP_PL_1	U_AP_PL_2	R_AP_PU_4	
PU_AP_PL_1	U_AP_PL_3	R_AP_PU_5	
I_PU	L_AP_PL_2	W_AP_PU_Z	
I_PL	L_AP_PL_3	W_AP_PU_NZ_1	
SI_E_PL	PU_AP_PL_2	W_AP_PU_NZ_2	
SI_T_PL	PU_AP_PL_3	W_AP_PU_NZ_3	
SI_U_PL		W_AP_PU_NZ_4	
		PU_AP	
		L_AP_PU_1	
		L_AP_PU_3	

Permalocked_APZ.txt	Permalocked_APNZ.txt	PermaUnlocked.txt	PermaLocked_L.txt
		PU_AP_PU_1	
		PU_AP_PU_2	
		PU_AP_PU_3	
		SI_E_PU	
		SI_T_PU	
		SI_U_PU	
		PU_U_PU	
		PU_T	

Table 6-8 SQ-EPC Test Cases

sq_epc_test.txt				
Test cases	Target	Action	Pointer	length
SQ_E_S0_1	S0	0	32	96
SQ_E_S0_2	S0	1	34	64
SQ_E_S0_3	S0	10	37	48
SQ_E_S0_4	S0	11	43	24
SQ_E_S0_5	S0	100	55	12
SQ_E_S0_6	S0	101	79	6
SQ_E_S0_7	S0	110	95	3
SQ_E_S0_8	S0	111	127	1
SQ_E_S1_1	S1	0	117	1
SQ_E_S1_2	S1	1	97	3
SQ_E_S1_3	S1	10	83	6
SQ_E_S1_4	S1	11	64	12
SQ_E_S1_5	S1	100	46	24
SQ_E_S1_6	S1	101	42	48
SQ_E_S1_7	S1	110	38	64
SQ_E_S1_8	S1	111	32	96
SQ_E_S2_1	S2	0	32	96
SQ_E_S2_2	S2	1	32	95
SQ_E_S2_3	S2	10	44	48
SQ_E_S2_4	S2	11	56	30
SQ_E_S2_5	S2	100	67	20
SQ_E_S2_6	S2	101	75	6
SQ_E_S2_7	S2	110	127	0
SQ_E_S2_8	S2	111	126	1
SQ_E_S3_1	S3	0	124	1
SQ_E_S3_2	S3	1	102	3
SQ_E_S3_3	S3	10	87	6
SQ_E_S3_4	S3	11	59	12
SQ_E_S3_5	S3	100	45	24

sq_epc_test.txt				
SQ_E_S3_6	S3	101	41	48
SQ_E_S3_7	S3	110	36	64
SQ_E_S3_8	S3	111	32	96
SQ_E_SL_1	SL	0	127	2
SQ_E_SL_2	SL	1	63	64
SQ_E_SL_3	SL	10	127	1
SQ_E_SL_4	SL	11	63	0
SQ_E_SL_5	SL	100	58	12
SQ_E_SL_6	SL	101	81	16
SQ_E_SL_7	SL	110	32	0
SQ_E_SL_8	SL	111	124	3
SQ_E_SL_9	SL	0	32	96
SQ_E_SL_10	SL	1	32	95
SQ_E_SL_11	SL	10	44	48
SQ_E_SL_12	SL	11	56	30
SQ_E_SL_13	SL	100	67	20
SQ_E_SL_14	SL	101	75	6
SQ_E_SL_15	SL	110	127	0
SQ_E_SL_16	SL	111	126	1

Table 6-9 SQ-TID Test Cases

sq_tid_test.txt				
Test cases	Target	Action	Pointer	length
SQ_T_S0_1	S0	0	32	Full
SQ_T_S0_2	S0	1	32	Half
SQ_T_S1_1	S1	10	32	Full
SQ_T_S1_2	S1	11	32	Half
SQ_T_S2_1	S2	100	32	Full
SQ_T_S2_2	S2	101	32	Half
SQ_T_S3_1	S3	110	32	Full
SQ_T_S3_2	S3	111	32	Half
SQ_T_SL_1	SL	0	32	Full
SQ_T_SL_2	SL	1	32	Half

Table 6-10 SQ-User Test Cases

sq_user_test.txt				
Test cases	Target	Action	Pointer	length
SQ_U_S0_1	S0	0	32	Full
SQ_U_S0_2	S0	1	32	Half
SQ_U_S1_1	S1	10	32	Full
SQ_U_S1_2	S1	11	32	Half

sq_user_test.txt				
SQ_U_S2_1	S2	100	32	Full
SQ_U_S2_2	S2	101	32	Half
SQ_U_S3_1	S3	110	32	Full
SQ_U_S3_2	S3	111	32	Half
SQ_U_SL_1	SL	0	32	Full
SQ_U_SL_2	SL	1	32	Half

Table 6-11 Applications (Annex N) - User Test Cases

Appl_EAS.txt	Appl_CE.txt	Tag Alt_Core. Txt	Tag Alt_Challenge. Txt	Tag Alt_Authenticate. Txt	Tag Alt_Full Txt
Test Cases	Test Cases	Test Cases	Test Cases	Test Cases	Test Cases
U_1	BP_1	U_1	U_1	U_1	U_1
A_1	A_1	A_1	A_1	A_1	A_1
	FO_1		C_1	AU_1	C_1
					BW_1
					AU_1
					SC_1
					KU_1

7 Scripting Language for Automated Interoperability Testing (Normative)

7.1 Script Language Syntax

The Reader vendor should provide a PC application that runs a Script File and produces an Output File. This application is invoked using the following syntax, where "InteropG2V2" is the call to the application and input file is the Script filename and output file is the Output filename.

InteropG2V2 <input file> <output file>

The input file (Script File) is a text file comprised of directives from a scripting language described below. The Script File shall be read by the application and each line executed in order. Results shall be written both to the screen and to the output file (Output File) for documentation and as inputs to the software manager program for parsing and report generation.

The scripting language is intended to be as simple as possible, both for the operator (in terms of being able to create powerful tests using simple building blocks) and for the application (in terms of being able to parse and understand the directives). The scripting language is line-oriented, meaning that individual directives within the testing language are separated by carriage returns. Blank lines are allowed and cause no actions to occur. Comments are indicated by a leading '#' character with the characters that follow having no operational influence. The comment character can occur anywhere on the line. Examples:

This is a line with nothing but a comment

Disconnect # This is a comment coming after a sample directive

It is advised to use informative comments when creating the testing scripts. All comments, blank lines, and directives are echoed to the output file. All comments shall have a Test Case ID embedded in each comment line.

The result of each directive is printed to the screen and the output file whose name is specified. Each directive can individually either succeed or fail. For example, a write command may fail due to the memory being locked. Another means of failure is a mismatch with an expected data result. The application shall indicate failure for any directive where the actual pass/fail result mismatches the expectation. All directives must match their expected result for overall success to be declared for a Script File. The application shall declare "SUCCESS" at the end of the Output File if all directives matched expectation. "FAILURE" is declared if any one directive had an unexpected result.

The core functionality of the testing language comes from its interpretation of a number of test directives. The list of supported directives and their syntax is shown below. If the application encounters an out of order or unrecognisable directive (lack of necessary arguments, presence of invalid arguments, attempting to perform RFID operations before connecting to a reader, etc.), then the directive is ignored and a diagnostic message describing the error is printed to the screen and output file. The following is the full list of supported directives. Parameters in <..> are mandatory, those in [..] are optional.

Core IOP Directives

Directive	Parm	Parm	Parm	Parm	Parm	Parm
Inventory	<expect>	<n>	[mask]	[location]		
Read	<expect>	<data>	<location>	[<offset.,<length>]	[password]	
Write	<expect>	<data>	<location>	[<offset.,<length>]	[password]	
WrRd	<expect>	<data>	<location>	[<offset.,<length>]	[password]	
Lock	<expect>	<location>	[password]			
UnLock	<expect>	<location>	[password]			
PermLock	<expect>	<location>	[password]			
PermUnLock	<expect>	<location>	[password]			
Kill	<expect>	[Kpassword				
SQ	<expect>	<location>	<epclength>	<target>	<action>	<pointer>
	<length>	<mask>	<truncate>	<session>	<sel>	<qtarget>

Inventory <expect> <n> [mask] [location]

Description:

This directive attempts to inventory tags that are in the field of view of the reader using repetitive inventory rounds. The option exists to select a sub-population of tags prior to inventory. Selection occurs based on an optional mask parameter field. This directive will succeed if the reader is able to inventory n tags where the value n is specified as a parameter; and will fail otherwise.

Parameters:

expect

Must be equal to "pass" or "fail", depending upon whether this directive is expected to succeed or not.

n

The number of unique EPCs that are expected to be found by the reader. If, for example, 10 tags are in the field of view of the reader and an inventory directive is executed without selection, then 10 tags should be found and the directive will pass if n=10 was specified. If a sub-population is targeted using the mask option, then a lower n value may be specified even though 10 total tags are in the field of view.

mask

A hexadecimal string that the EPC must match for the select criteria to apply. String corresponds to a Select mask starting at the EPC MSB. If not specified, default is no Select mask, that is, complete population is inventoried.

location

Use "epc" for the EPC memory bank, "tid" for the TID memory bank, and "user" for the user memory bank (for tags with no files, FN_0). Use 1,2,3,…1023 for tags that have implemented file system, the number indicative of the file number.

If not specified, default is epc.

Use "challenge" only for the command to challenge a tag population for subsequent authentication. This parameter is only used for testing Tag Alteration Challenge and Tag Alteration Full as part of Application IOP test category.

Examples:

Inventory pass 1

Inventory fail 3 F3C5 user

Inventory fail 3 F3C5 2

Inventory pass 1 F3C5 challenge

Result:

Inventoried <count> unique tags.

The count is the number of unique EPCs found.

EPC result: <epc_value> <:count> (see following example)

30035A0001B4F449A720CD20 :33

30035A0001B4F449A720CD21 :30

30035A0001B4F449A720CD22 :25

A hexadecimal list of EPCs found followed by the decimal number of times each was inventoried.

Inventory <result>.

The result is "SUCCESSFUL" if the directive matched the expectation, or "FAILED" if the directive mismatched the expectation. The number of unique tags found is compared to the number of expected tags to determine success/failure.

Read <expect> <data> <location>[,<offset>,<length>] [password]

Description:

This directive attempts to read a particular memory location in the tag under test. The memory bank is specified in the location field. Optional offset and length parameters specify the start and length of data to read if a partial memory read is desired. If this field is omitted, the complete memory will be read. The data field contains the data expected to be return from the read. A dash in this field eliminates the requirement for a match with expected data to achieve success. The password field is used if a password is required to perform the operation. If the password is omitted, memory access will be attempted from the open state. This directive will succeed if the data can be read and it matches the expected data; and will fail otherwise.

Parameters:

expect

Must be equal to "pass" or "fail", depending upon whether this directive is expected to succeed or not.

data

A hexadecimal string that the read data will be matched against, or "-" to indicate that no matching should be performed.

location

Use "kpass" for the kill password, "apass" for the access password, "epc" for the EPC memory bank, "tid" for the TID memory bank, and "user" for the user memory bank (for tags with no files, FN_0). Use 1,2,3,…1023 for tags that have implemented file system, the number indicative of the file number.

offset

Decimal word offset (word is 16 bits) to start of data to written. If omitted, the starting location of the memory bank or file is used.

length

Decimal word length to read. If omitted, the all the words in the memory bank are read.

password

A hexadecimal string that will be used by the reader to perform a tag access from the secured state. If the password is omitted the tag access is performed from the open state.

Examples:

Read pass - tid

Read fail 1111 kpass,1,1 33334444

Result:

Read data: <data>

The data is a hexadecimal string of the data that was read from tag memory, or "" if no data could be read.

Read <result>.

The result is "SUCCESSFUL" if the directive matched the expectation, or "FAILED" if the directive mismatched the expectation. If data is specified it is compared to the read result to determine success/failure.

Write <expect> <data> <location>[,<offset>] [password]

Description:

This directive attempts to write a particular memory location in the tag under test. The memory bank is specified in the location field. An optional offset parameter specifies the starting position in the bank to write if a partial memory write is desired. If this field is omitted, the complete memory will be written. The data field contains the data to be written. The password field is used if a password is required to perform the operation. If the password is omitted, memory access will be attempted from the open state. This directive will succeed if the tag reports a successful write and the read results matches data if data is specified; and will fail otherwise.

Parameters:

expect

Must be equal to "pass" or "fail", depending upon whether this directive is expected to succeed or not.

data

A hexadecimal string that represents the data to be written. If it is shorter than the length of remaining memory then the higher-numbered rows will be left unmodified.

location

Use "kpass" for the kill password, "apass" for the access password, "epc" for the EPC memory bank, "crc16" for the CRC-16 memory bank, "tid" for the TID memory bank, and "user" for the user memory bank (for tags with no files, FN_0). Use 1,2,3,…1023 for tags that have implemented file system, the number indicative of the file number.

offset

Decimal word offset (word is 16 bits) to start of data to written. If omitted, the starting location of the memory bank is used.

password

A hexadecimal string that will be used by the reader to perform a tag access from the secured state. If the password is omitted the tag access is performed from the open state.

Examples:

Write pass 11112222 kpass

Write fail 1111 tid,1 33334444

Result:

Write result: <data>

The data is "1" if the write operation was successful, or "0" if it was not successful.

Write <result>.

The result is "SUCCESSFUL" if the directive matched the expectation, or "FAILED" if the directive mismatched the expectation.

WrRd <expect> <data> <location>[,<offset>] [password]

Description:

This directive attempts to write a particular memory location of the tag under test then read the same location to verify a successful write. The memory bank is specified in the location field. An optional offset parameter specifies the starting position in the bank to write if a partial memory write is desired. If this field is omitted, the complete memory will be written. The data field contains the data to be written. The password field is used if a password is required to perform the operation. If the password is omitted, memory access will be attempted from the open state. This directive will succeed if the tag reports a successful write and the read data matches what was written; and will fail otherwise.

Parameters:

expect

Must be equal to "pass" or "fail", depending upon whether this directive is expected to succeed or not.

data

A hexadecimal string that represents the data to be written. If it is shorter than the length of remaining memory then the higher-numbered rows will be left unmodified.

location

Use "kpass" for the kill password, "apass" for the access password, "epc" for the EPC memory bank, "tid" for the TID memory bank, and "user" for the user memory bank (for tags with no files, FN_0). Use 1,2,3,…1023 for tags that have implemented file system, the number indicative of the file number.

offset

Decimal word offset (word is 16 bits) to start of data to written. If omitted, the starting location of the memory bank is used.

password

A hexadecimal string that will be used by the reader to perform a tag access from the secured state. If the password is omitted the tag access is performed from the open state.

Examples:

WrRd pass 11112222 kpass

WrRd fail 1111 tid,1 33334444

Result:

WrRd write result: <data>

The data is "1" if the write operation was successful, or "0" if it was not successful.

WrRd read result: <data>

The data is a hexadecimal string of the data that was read from tag memory, or "" if no data could be read. Note that this statement is omitted if the read operation is not done. This happens when the write operation mismatches the directive expectation.

WrRd <result>

The result is "SUCCESSFUL" if the directive matched the expectation, or "FAILED" if the directive mismatched the expectation. Data is compared to the read result to determine success/failure.

Lock <expect> <location> [password]

UnLock <expect> <location> [password]

PermLock <expect> <location> [password]

PermUnLock <expect> <location> [password]

Description:

All four of these directives are used to change the lock state of the tag under test. These directives set and clear the lock bit, and they set the permalock bit in the location field of Table 6.50 in Gen 2 protocol specification. For example, if the tag memory is in the unlocked state (lock and permalock bits both zero) a PermLock directive will set the permalock bit to one and put the memory in a perma-unlocked state. Likewise, the tag memory must set the lock bit by issuing a Lock directive either prior or post to issuing a PermLock directive to put the tag memory in a permalock state. The PermUnLock directive attempts to deassert the permalock bit. The password field is used if an access password is required to perform the operation. If the password is omitted, memory access will be attempted from the open state. This directive will succeed if the tag reports a successful lock or unlock; and will fail otherwise.

Parameters:

expect

Must be equal to "pass" or "fail", depending upon whether this directive is expected to succeed or not.

location

Use "kpass" for the kill password, "apass" for the access password, "epc" for the EPC memory bank, "tid" for the TID memory bank, and "user" for the user memory bank(for tags with no files, FN_0). Use 1,2,3,…1023 for tags that have implemented file system, the number indicative of the file number.

password

A hexadecimal string that will be used by the reader to perform a tag access from the secured state. If the password is omitted the tag access is performed from the open state.

Examples:

Lock pass user 33334444

UnLock fail epc

PermLock fail user

PermUnLock pass epc 33334444

Result:

Lock result: <data>

UnLock result: <data>

PermLock result: <data>

PermUnLock result: <data>

The data is "1" if the lock operation was successful, or "0" if it was not successful.

Lock <result>.

UnLock <result>.

PermLock <result>.

PermUnLock <result>.

The result is "SUCCESSFUL" if the directive matched the expectation, or "FAILED" if the directive mismatched the expectation.

Kill <expect> <kpassword>

Description:

This directive attempts to kill the tag under test using the kill password specified in the password field. The directive succeeds if the tag reports a successful kill operation; and will fail otherwise.

Parameters:

expect

Must be equal to "pass" or "fail", depending upon whether this directive is expected to succeed or not.

kpassword

A hexadecimal string that specifies the kill password to be used for the kill operation.

Examples:

Kill pass FFFFFFFF

Result:

Kill result: <data>

The data is "1" if the kill operation was successful, or "0" if it was not successful.

Kill <result>.

The result is "SUCCESSFUL" if the directive matched the expectation, or "FAILED" if the directive mismatched the expectation.

SQ <expect> <location> <epclength> <starget> <action> <pointer> <length> <mask> <truncate> <session> <sel> <qtarget>

Description:

The Select/Query directive causes the reader to send a Select command followed immediately by a Query command to the tag. The parameter list contains many of the Select and Query parameters specified in the Gen 2 protocol specification. Parameters that are not included are determined by the reader mode which is set in the configuration directive. The Q value parameter in the Query command shall be defaulted to zero to assure an immediate response from the single tag that is being interrogated.

Parameters:

expect

Must be equal to "pass" or "fail", depending upon whether this directive is expected to succeed or not. Pass should be specified when a valid EPC is expected to be returned by the tag.

location

Use "epc" for the EPC memory bank, "tid" for the TID memory bank, and "user" for the user memory bank(for tags with no files, FN_0). Use 1,2,3,…1023 for tags that have implemented file system, the number indicative of the file number.

epclength

Decimal integer indicating the number of EPC bits supported by the tag-under-test. A typical value is 96 bits.

starget

The Select target parameter specifies which flag to modify if the select mask matches the tag memory value. Use "s0" for the inventoried S0 flag, "s1" for the inventoried S1 flag, "s2" for the inventoried S2 flag, "s3" for the inventoried S3 flag, and "sl" for the SL flag.

action

Action is a three digit binary number as defined in Table 6.30 of the Gen 2 protocol specification that defines the action the targeted flag takes when the mask matches and mismatches the value in the tag memory.

pointer

Decimal integer that specifies the memory offset at which the select mask is applied. The pointer is not in the extensible bit vector format used in the Gen 2 Select command.

length

Decimal integer that specifies the length of the select mask in bits, with the length being left justified for selecting the portion of the mask. For mask lengths that are not multiples of 4, the mask used will be padded with zeros for its remainder least significant bits. For example, if the mask is 7F, and the length is 7, the mask used will be 7E.

mask

A hexadecimal string indicating the bit pattern to be compared with the tag memory for purposes of selecting a sub-population of tags. If the mask length is not a multiple of four then the mask will be left justified with trailing zeros in the unused LSB's.

truncate

Use "1" to enable a truncated reply from tag and "0" for an un-truncated response.

session

The Query session parameter specifies which flag to use for inventorying. Use "s0" for the S0 flag, "s1" for the S1 flag, "s2" for the S2 flag, and "s3" for the S3 flag.

sel

A two digit binary number specifying whether the selected, unselected or all tags respond to a Query. See Table 6.32 in Gen 2 protocol specification for details.

qtarget

The Query target parameter. Use "A" to choose tags in A state to participate in inventory and "B" to choose B state tags for participation.

Examples:

SQ pass epc sl 000 45 30 B345FE14 0 s0 00 A

SQ fail user s0 000 32 64 123AB45FA125BC12 0 s0 00 B

Result:

SQ result: <data>

The data is "1" if the tag responded with a valid EPC value, or "0" if there was no tag response.

EPC result: <epc_value> (see following example)

30035A0001B4F449A720CD22

The returned EPC if data=1; no output if data=0

SQ <result>.

The result is "SUCCESSFUL" if the directive matched the expectation, or "FAILED" if the directive mismatched the expectation.

Connect <address> <domain> <mode><version>

Description:

This directive attempts to connect to the reader. The directive should be used before attempting any testing directives, and should not be used when already connected to a reader. The target reader is specified by the address parameter. The domain and mode parameters specify configuration information for that reader. The version specifies the version of the specification.

Parameters:

address

The address of the reader to connect to. This can either be an IP address or, if DNS resolution is available, the name of the reader.

domain

Use "FCC" for connecting to a reader operating within the FCC domain, "ETSI" for connecting to a reader operating within the ETSI domain, "JPN" for connecting to a reader operating within the Japanese domain, and "CHN" for connecting to a reader operating within the Chinese domain.

mode

An positive integer mode number that uniquely specifies the modulation type, Tari, PIE, LF, M, DR, and TRext used by the reader and commanded of the tag.

version

Use "2.0.0" or "1.1.0"

Examples:

Connect speedway0111 FCC 2 1.1.0

Connect 192.168.10.51 ETSI 0 2.0.0

Disconnect

Description:

This directive attempts to disconnect a connected reader. The directive should be used for reconnecting to a different reader in the midst of a test. It is not necessary to issue this directive at the end of the test.

Parameters:

Example:

Disconnect

Power <power>

Description:

This directive attempts to set the reader transmit power level as measured at the RF connector.

Parameters:

power

A floating-point value in the range between 15.0 and 30.0, inclusive. Units are dBm. The reader shall set the power level as closely as possible to the commanded value.

Example:

Power 15.0

Antenna <m>

Description:

This directive is used to select the active transmit antenna port. Only one transmit antenna port shall be active at a time during any interoperability testing.

Parameters:

m

An integer greater than or equal to 1.

Example:

Antenna 2

Frequency <frequency>

Description:

This directive attempts to set the operating frequency to be used by the reader. This directive applies only to readers designed for operation in a region allowing fixed frequency operation.

Parameters:

frequency

A floating-point value in the range between Fmin and Fmax representing the minimum and maximum channel frequencies for the particular regulatory region. Units are MHz. For ETSI the values are 865.7 and 867.5.

Example:

Frequency 866.3

Application IOP (Annex N) Directives

Directive	Parm	Parm	Parm	Parm	Parm	Parm
Untraceable	<expect>	<U>	<epc>	<TID>	<user>	<range>
Access	<expect>	<pswd>				
Challenge	<expect>	<CSI>	<message>	<length>		
Authenticate	<expect>	<CSI>	<message>	<length>		
BlockWrite	<expect>	<data>	<MemBank>	<WordPtr,>	<WordCount>]	
SecureComm	<expect>	<message>	<length>			
BlockPermalock	<expect>	<MemBank>	<BlockPtr>	<BlockRange>	<Mask>	
FileOpen	<expect>	<FileNum>				
ReadBuffer	<expect>	<data>	<WordPtr>	<BitCount>		

The contents and structure of parameters for each directive are the same as in Core IOP.

Table 4-1 shows which directives are used in each Application IOP.

Extended IOP (Optional commands) Directives

Note: Directives are also used as part of Application IOP

Directive	Parm	Parm	Parm	Parm	Parm	Parm
Access	<expect>	<pswd>				
BlockWrite	<expect>	<data>	<MemBank>	<WordPtr,>	<WordCount>]	
BlockErase	<expect>	<MemBank>	<WordPtr,>	<WordCount>]		
BlockPermalock	<expect>	<MemBank>	<BlockPtr>	<BlockRange>	<Mask>	
Untraceable	<expect>	<U>	<epc>	<TID>	<user>	<range>
FileOpen	<expect>	<FileNum>				
FileList	<expect>	<FileNum>	<AddlFile>			
FileSetup	<expect>	<FileType>	<FileSize>			
FilePrivilege	<expect>	<Action>	<KeyID>	<Privilege>		
Authenticate	<expect>	<CSI>	<msg. variable>	<length>		
AuthComm	<expect>	<message>	<length>			
SecureComm	<expect>	<message>	<length>			
KeyUpdate	<expect>	<keyID>	<message>	<length>		
TagPrivilege	<expect>	<Action>	<Target>	<KeyID>	<Privilege>	
ReadBuffer	<expect>	<data>	<WordPtr>	<BitCount>		

The contents and structure of parameters for each directive are the same as in Core IOP.

7.2 Interoperability Script Input File Example

The following Script File illustrates the use of a commented string of directives to test secured access of tag memory.

```
Connect speedway0022 FCC 2 1.1.0
Power 27.0

# Read kill and access passwords.
Read pass - kpass
Read pass - apass

# Test partial writing of kill password.
WrRd pass 00000000 kpass
WrRd pass FFFF kpass,1
Read pass 0000FFFF kpass

# Write kill and access passwords.
WrRd pass AAAAAAAA kpass
WrRd pass BBBBBBBB apass
```

```
# Lock kill and access passwords.
Lock pass kpass BBBBBBBB
Lock pass apass BBBBBBBB

# Read kill and access passwords.
Read pass AAAAAAAA kpass BBBBBBBB
Read pass BBBBBBBB apass BBBBBBBB

# Fail to kill tag with wrong password.
Kill fail 11111111

# Unlock kill and access passwords.
UnLock pass kpass BBBBBBBB
UnLock pass apass BBBBBBBB

# Read EPC.
Read pass - epc

# Write EPC from open state.
WrRd pass 1234567890ABCDEF12345678 epc

# Lock EPC
Lock pass epc BBBBBBBB

# Write EPC from secured state.
WrRd pass 111122223333444455556666 epc BBBBBBBB

# Unlock EPC.
UnLock pass epc BBBBBBBB

# Reset kill and access passwords.
WrRd pass 00000000 kpass
WrRd pass 00000000 apass

Disconnect
```

7.3 Interoperability Script Output File Example

The following Output file shows the output from the Script File example from the above example.

```
Connect speedway0022 FCC 2 1.1.0
    Connected to reader speedway0022, type 0, mode 2, air interface version 1.1.0.

Power 27.0
    Setting power to 27.0 dBm

# Read kill and access passwords.
Read pass - kpass
    Read data: 00000000
    Read SUCCESSFUL.
Read pass - apass
    Read data: 00000000
    Read SUCCESSFUL.

# Test partial writing of kill password.
WrRd pass 00000000 kpass
    WrRd write result: 1
    WrRd read result: 00000000
    WrRd SUCCESSFUL.
WrRd pass FFFF kpass,1
    WrRd write result: 1
    WrRd read result: FFFF
    WrRd SUCCESSFUL.
Read pass 0000FFFF kpass
    Read data: 0000FFFF
```



```
    Read SUCCESSFUL.

# Write kill and access passwords.
WrRd pass AAAAAAAA kpass
    WrRd write result: 1
    WrRd read result: AAAAAAAA
    WrRd SUCCESSFUL.
WrRd pass BBBBBBBB apass
    WrRd write result: 1
    WrRd read result: BBBBBBBB
    WrRd SUCCESSFUL.

# Lock kill and access passwords.
Lock pass kpass BBBBBBBB
    Lock result: 1
    Lock SUCCESSFUL.
Lock pass apass BBBBBBBB
    Lock result: 1
    Lock SUCCESSFUL.

# Read kill and access passwords.
Read pass AAAAAAAA kpass BBBBBBBB
    Read data: AAAAAAAA
    Read SUCCESSFUL.
Read pass BBBBBBBB apass BBBBBBBB
    Read data: BBBBBBBB
    Read SUCCESSFUL.

# Fail to kill tag with wrong password.
Kill fail 11111111
    Kill result: 0
    Kill SUCCESSFUL.

# Unlock kill and access passwords.
UnLock pass kpass BBBBBBBB
    UnLock result: 1
    UnLock SUCCESSFUL.
UnLock pass apass BBBBBBBB
    UnLock result: 1
    UnLock SUCCESSFUL.

# Read EPC.
Read pass - epc
    Read data: 11112222333344445555666660000
    Read SUCCESSFUL.

# Write EPC from open state.
WrRd pass 1234567890ABCDEF12345678 epc
    WrRd write result: 1
    WrRd read result: 1234567890ABCDEF12345678
    WrRd SUCCESSFUL.

# Lock EPC
Lock pass epc BBBBBBBB
    Lock result: 1
    Lock SUCCESSFUL.

# Write EPC from secured state.
WrRd pass 111122223333444455556666 epc BBBBBBBB
    WrRd write result: 1
    WrRd read result: 111122223333444455556666
    WrRd SUCCESSFUL.

# Unlock EPC.
UnLock pass epc BBBBBBBB
    UnLock result: 1
```



```
    UnLock SUCCESSFUL.

# Reset kill and access passwords.
WrRd pass 00000000 kpass
    WrRd write result: 1
    WrRd read result: 00000000
    WrRd SUCCESSFUL.
WrRd pass 00000000 apass
    WrRd write result: 1
    WrRd read result: 00000000
    WrRd SUCCESSFUL.

Disconnect
    Disconnected.

    OVERALL RESULT: SUCCESS
```


ANNEX A (Normative) IS and IXIT specification

A.1 Scope

The present document provides the Implementation Statement (IS) *pro forma* for the radio-frequency identification (RFID) system operating in the 860 MHz – 960 MHz frequency range defined in EPCglobal Class-1 Generation-2 UHF RFID Protocol V.2.0.0 in compliance with the relevant requirements, and in accordance with the relevant guidance given in ISO/IEC 9646.

A.2 References

1. EPCglobal Class-1 Generation-2 UHF RFID Protocol V.2.0.0
2. ISO/IEC 9646-1: "Information technology - Open systems interconnection - Conformance testing methodology and framework – Part 1: General concepts".
3. ISO/IEC 9646-7: "Information technology - Open systems interconnection - Conformance testing methodology and framework – Part 7: Implementation Conformance Statements".
4. ETSI TS 102 237-1 v 4.1.1 (2003-12): Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Interoperability test methods and approaches; Part 1: Generic approach to interoperability testing.

A.3 Definitions

In particular, the following terms and definitions apply:

Implementation Statement (IS): It is a checklist of the capabilities/functionalities supported by the Device Under Test (DUT). IS is used to select and parameterize test cases and as an indicator for basic interoperability between different products.

IS *proforma*: A document, in the form of a questionnaire, which when completed for an implementation or system becomes an IS.

Implementation eXtra Information for Testing (IXIT): It contains additional information (e.g., specific addresses, timer values, etc.) necessary for testing.

A.4 Abbreviations

IS Implementation Statement

DUT Device Under Test

A.5 Conformance to this IS proforma specification

If it claims to conform to the present document, the actual IS proforma to be filled in by a supplier shall be technically equivalent to the text of the IS proforma given in annex A, and shall preserve the numbering/naming and ordering of the proforma items.

An IS which conforms to the present document shall be a conforming IS proforma completed in accordance with the guidance for completion given in clause A.6.

IS proforma for EPCglobal Class-1 Generation-2 UHF RFID Protocol V.2.0.0. Interoperability Test Cases.

A.6 Guidance for completing the IS proforma

A.6.1 Purposes and structure

The purpose of this IS proforma is to provide a mechanism whereby a supplier of an implementation of the requirements defined in the specification "EPCglobal Class-1 Generation-2 UHF RFID Protocol" may provide information about the implementation in a standardized manner.

The IS proforma is subdivided into clauses for the following categories of information:

- Guidance for completing the IS proforma;
- Identification of the implementation;
- Identification of the <reference specification type>;
- Global statement of conformance;
- Instructions for completing the IS proforma;
- Identification of the implementation;
- Identification of the protocol;
- Global statement of conformance;
- Roles;
- Major capabilities;
- Timers;
- Extra information for testing.

A.6.2 Abbreviations and conventions

Item column

The item column contains a number which identifies the item in the table.

Item description column

The item description column describes in free text each respective item (e.g. parameters, timers, etc.). It implicitly means "is <item description> supported by the implementation?".

Document reference column

The document reference column makes reference to EPCglobal Class-1 Generation-2 UHF RFID Protocol V.2.0.0 except where explicitly stated otherwise.

IS reference column

The IS reference contains the identifier of a particular item. It is used in the selection criteria of each test purpose.

Status column

The following notations, defined in ISO/IEC 9646-7, are used for the status column:

m mandatory - the capability is required to be supported.

o optional - the capability may be supported or not.

n/a not applicable - in the given context, it is impossible to use the capability.

x prohibited (excluded) - there is a requirement not to use this capability in the given context.

o.i qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer which identifies an unique group of related optional items and the logic of their selection which is defined immediately following the table.

Support column

The support column shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7, are used for the support column:

Y or y supported by the implementation.

N or nnot supported by the implementation.

N/A, n/a or - no answer required (allowed only if the status is n/a, directly or after evaluation of a conditional status).

Values allowed column

The values allowed column contains the type, the list, the range, or the length of values allowed. The following notations are used:

Range of values: <min value> .. <max value>

Example: 5 .. 20

Values supported column

The values supported column shall be filled in by the supplier of the implementation. In this column, the values or the ranges of values supported by the implementation shall be indicated.

A.6.3 Instructions for completing the IS proforma

The supplier of the implementation shall complete the IS proforma in each of the spaces provided. In particular, an explicit answer shall be entered, in each of the support or supported column boxes provided, using the notation described in clause

If necessary, the supplier may provide additional comments in space at the bottom of the tables or separately.

More detailed instructions are given at the beginning of the different clauses of the IS proforma.

A.6.4 Identification of the implementation

Identification of the Device Under Test (DUT) should be filled in so as to provide as much detail as possible regarding version numbers and configuration options.

The product supplier information and client information should both be filled in if they are different.

A person who can answer queries regarding information supplied in the IS should be named as the contact person.

Date of the statement

.....

Device Under Test (DUT) identification

DUT name:

.....

.....

DUT version:

.....

Hardware configuration:

.....

.....

.....

Operating system:

.....

Product supplier

Name:

.....

Address:

.....

.....

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

.....

Client (if different from product supplier)

Name:

.....

Address:

.....

.....

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

IS contact person

(A person to contact if there are any queries concerning the content of the IS)

Name:

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

.....

Identification of the specification

This IS proforma applies to the Interoperability test cases of the following standard: EPCglobal Class-1 Generation-2 UHF RFID Protocol V.2.0.0.

Global statement of conformance

Are all mandatory capabilities implemented? (Yes/No)

NOTE: Answering "No" to this question indicates non-conformance to the specification. Non-supported mandatory capabilities are to be identified in the IS, with an explanation of why the implementation is non-conforming, on pages attached to the IS proforma.

A.6.5 Roles

Table A-1 Roles

Item	Role	Status	Support
1	Interrogator	o.1	
2	Tag	o.1	
3	Other(specify)	o.1	

Comments:

A.6.5.1 Interrogator Role

Table A-2 Basic operations and capabilities for managing tag populations

Item	Operation	Status	Support
1	Select	m	
2	Inventory	m	
3	Access	m	
4	Interrogators supports 4 sessions (denoted S0, S1, S2, S3)	m	

Comments:

Table A-3 Commands Supported

Item	Command	Status	Support
1	Select	m	
2	Challenge	o	
3	Query	m	
4	QueryAdjust	m	
5	QueryRep	m	
6	ACK	m	
7	NAK	m	
8	Req_RN	m	
9	Read	m	
10	Write	m	
11	Kill	m	

Item	Command	Status	Support
12	Lock	m	
13	Access	o	
14	BlockWrite	o	
15	BlockErase	o	
16	BlockPermalock	o	
17	Authenticate	o	
18	AuthComm	o	
19	SecureComm	o	
20	KeyUpdate	o	
21	TagPrivilege	o	
22	ReadBuffer	o	
23	Untraceable	o	
24	FileOpen	o	
25	FileList	o	
26	FilePrivilege	o	
27	FileSetup	o	

Comments:

A.6.5.2 Tag Role

Table A-4 Memory banks supported

Item	Memory bank	Status	Support
1	Reserved memory	m	
2	EPC memory	m	
3	TID memory	m	
4	User memory	o	
5	File System	o	

Comments:

Provide details of File System Implementation

Table A-5 Stored passwords in Reserved memory bank

Item	Password	Status	Support
1	Kill password	o	
2	Access password	o	

Comments:

Table A-6 Stored data in EPC memory bank

Item	EPC Data	Status	Support
1	CRC-16	m	
2	PC (Protocol control)	m	
3	Object identifier code	m	
4	XPC_W1	o	
5	XPC_W2	o	

Comments:

Table A-7 Object identifier type

Item	Object identifier	Status	Support
1	EPC (EPCglobal members)	o.1	
2	Other object identifier code	o.1	

Comments:

c.1: It is mandatory to support one of these items.

Table A-8 Stored data in TID memory bank

Item	TID Data	Status	Support
1	8-bit ISO/IEC 15963 allocation class identifier	m	
2	9-bit tag mask-designer identifier	c.1	
3	12-bit tag model number	c.1	
4	XTID	o	
5	TID memory location 00_h to $1F_h$ shall be permalocked at time of manufacture.	c.1	
6	Is the TID data stored above location $1F_h$ locked/permalocked/ permaunlocked	o	
7	Entire XTID shall be permalocked if XTID is implemented.	c.1	

Comments:

c.1: Mandatory for EPCglobal members

Table A-9 States

Item	States	Status	Support
1	Ready state	m	
2	Arbitrate state	m	
3	Reply state	m	
4	Acknowledged state	m	
5	Open state	m	
6	Secured state	m	
7	Killed state	m	

Comments:

Table A-10 Commands Supported

Item	Command	Status	Support
1	Select	m	
2	Challenge	o	
3	Query	m	
4	QueryAdjust	m	
5	QueryRep	m	
6	ACK	m	
7	NAK	m	
8	Req_RN	m	
9	Read	m	
10	Write	m	
11	Kill	m	
12	Lock	m	
13	Access	o	
14	BlockWrite	o	
15	BlockErase	o	
16	BlockPermalock	o	
17	Authenticate	o	
18	AuthComm	o	
19	SecureComm	o	
20	KeyUpdate	o	
21	TagPrivilege	o	
22	ReadBuffer	o	
23	Untraceable	o	
24	FileOpen	o	
25	FileList	o	
26	FilePrivilege	o	
27	FileSetup	o	

Comments:

Table A-11 Logical partitioning of the memory banks

Item	MemBank	Status	Support	Values	
				Allowed	Supported
1	Reserved memory bank	m		00	
2	EPC memory bank	m		01	
3	TID memory bank	m		10	
4	User memory bank	o		11	
5	File System	c.1		FileType	FileNumber
	File_1				
	File_N				

Comments:

c.1. Provide details on File System implementation.

Table A-12 Memory locations of stored data in each memory bank

Item	Data	Status	Support	Values	
				Allowed	Supported
1	Kill password in Reserved memory	c.1		00_h - $1F_h$	
2	Access password in Reserved memory	c.2		20_h - $3F_h$	
3	CRC-16 in EPC memory	m		00_h – $0F_h$	
4	Protocol Control (PC) in EPC memory	m		10_h – $1F_h$	
5	XI	m		16_h	c.3
6	Object identifier code in EPC memory	m		20_h – $20F_h$	
7	XPC_W1	o		210_h – $21F_h$	
8	XPC_W2	o		220_h – $22F_h$	
9	ISO/IEC 15963 allocation class identifier in TID memory	m		00_h - 07_h	
10	Other identifier information in TID memory	m		Above 07_h	
11	XTID (X) indicator	c.4		08_h	
12	Security (S) indicator	c.4		09_h	
13	File (F) indicator	c.4		$0A_h$	
14	Tag mask-designer identifier in TID memory	c.4		$0B_h$ – 13_h	
15	Tag model number in TID memory	c.4		14_h – $1F_h$	

Comments:

c.1: Mandatory for tags that implement the kill password

c.2: Mandatory for tags that implement the access password

c.3: declare how to compute XI=0

c.4: Mandatory for EPCglobal members

Table A-13 Cryptographic-Suite Indicators

Item	Assigning Authority (section 2.5 Spec)	Four most significant bits	Four least significant bits	Status	Support
1				c.1.	
2					

Comments

C.1 Mandatory for tags that implement file system.

Table A-14 Applications Implemented (per Annex N)

Item	Application	Status	Support	Values	
				Allowed	Supported
1	Alteration EAS				
2	Tag Alteration (Core)				
3	Tag Alteration (Challenge)				
4	Tag Alteration (Authenticate)				
5	Tag Alteration (Full)				
6	Consumer Electronics				

Comments

IXIT proforma for GS1 EPCglobal Class-1 Generation-2 UHF RFID Protocol V2.0.0. Interoperability Test Cases

A.7 Operating Parameters

Is the interrogator controlled by means of an external PC or terminal unit?
(Yes/No)

Does the interrogator notify (e.g. by means of messages) if the operation has been executed correctly?
(Yes/No)

Supply voltage:_____ _____

Supply current:_____ _____

Operating frequency range:_____

Implemented modulation(s) format in Interrogator-to-Tag communication:_____

Implemented modulation(s) format in Tag-to-Interrogator communication:_____

Implemented codification format in Tag-to-Interrogator communication:_____

Frequency sub-band:_____

Number of channels:_____ _____

Channel width:_____

Interoperability test mode description:

ANNEX B (Normative) Test Report

Report No.: **#report number#**

TEST NAME: **#test name#**

Product Name	:	#product name#
Trade Mark	:	#trade mark#
Product ID	:	IUT_PRODUCT_ID
Manufacturer	:	MANUFAC_COMPANY
Client	:	CLIENT_COMPANY
Standard(s)	:	GS1 EPCglobal Class-1 Generation-2 UHF Protocol V2.0.0

B.1 IDENTIFICATION SUMMARY

B.1.1 Test Campaign Report

Name:	
Address	
City:	
Postal code:	
Country:	
Telephone:	
Fax:	
URL:	
Contact person:	
Name:	
e-mail:	

B.1.2 Test Parameters for QE Interrogators Used for Verification

Reader name:				
Hardware Rev:				
Firmware version:				
Software version used to run TCs:				
Serial Number:		**General Characteristics**		
Frequency (MHz)	**Tari(us)**	**Modulation**	**Backscatter encoding**	**Data rate (Kbps)**

B.1.3 Test Parameters for Tag Under Testing

Tag Manufacturer	
Product Model Number	
IC Manufacturer	
IC Model Number	
Intended region of operation (All/NA/EU/JA)	
Intended Application	
EPC Size	
TID Size	
User memory and size (if present)	
Access control (access password)?	
Kill enabled (supports nonzero kill password)?	
Maximum RF Input Power (dBm)	

B.1.4 QE Tags Used for Verification

Competences and guarantees:

#LABRATORYNAME# is a testing laboratory competent to carry out the tests described in this report.

In order to assure the traceability to other national and international laboratories, **#LABRATORYNAME#** has a calibration and maintenance programme for its measuring equipment.

#LABRATORYNAME# guarantees the reliability of the data presented in this report, which is the result of measurements and tests performed to the item under test on the date and under the conditions stated on the report and is based on the knowledge and technical facilities available at **#LABRATORYNAME#** at the time of execution of the test.

#LABRATORYNAME# is liable to the client for the maintenance by its personnel of the confidentiality of all information related to the item under test and the results of the test.

B.1.5 Client

Name:	
V.A.T. :	
Address:	
City:	
Postal code:	
Country:	
Telephone:	
Contact person:	
Name:	
e-mail:	

B.1.6 Manufacturer

Name:	
V.A.T. :	
Address:	
City:	
Postal code:	
Country:	
Telephone:	
Contact person:	
Name:	
e-mail:	

B.1.7 Implementation Under Test

Product Name:	
Trademark:	
Product ID:	
Hw version:	c.1
Sw version:	c.1
Profiles supported:	c.1
Protocol Specification(s):	
IS:	SEE ANNEX A
IXIT:	SEE ANNEX B
Description of DUT:	
Sample method:	Samples undergoing test have been selected by: The Client

Comments: c.1 Applicable to interrogator under test.

c.2 Applicable to tag under test.

B.1.8 Test case iterations tested (Applicable to interrogator under test)

Frequency (MHz)	Tari(us)	Modulation	Backscatter encoding	Data rate (kbps)	DR	PIE

B.1.9 Testing Environment

Period of testing:	
Interoperability log reference:	SEE ANNEX C
Retention date for log reference:	5 years
Test Requested	Interoperability testing for **#DUT#.**

B.1.10 Test conditions:

NOMINAL

TEMPERATURE IN THE RANGE 18ºC TO 27 ºC	

B.1.11 Limits and reservations

The test results presented in this test report apply only to the device under test (DUT) for the functionality described in the relevant Implementation Statement (IS), as presented for test on the date(s) declared in section B.1.11 and configured as declared in the relevant Implementation eXtra Information for Testing (IXIT).

This test report does not constitute or imply, by its own, to be an approval of the product by Qualification Bodies, Certification Bodies or competent Authorities.

This document is only valid if complete; no partial reproduction can be made without written approval of the Test Laboratory.

This test report cannot be used partially or in full for publicity and/or promotional purposes without previous written approval of the Test Laboratory.

B.1.12 Record of agreement

The following samples were used for testing.

INTERNAL CONTROL NO.:	ELEMENT:	SERIAL NO.:	DATE OF RECEPTION:
xxxxx/13			
xxxxx/05			

B.2 Test Result Summary

B.2.1 QE Devices Interoperability

Was DUT Interoperable with QE devices?	

B.2.2 Test Campaign Report

The abbreviations used in the header row of the test campaign report tables are:

Applicable:	Indicates whether or not a test case has been selected for execution against DUTs identified in section B.1.11 according to the analysis of the information in the IS and IXIT for DUT.
Run:	Indicate whether or not the corresponding test case has been run to completion.
Verdict:	Records the verdict assigned to each test case run to completion. Following verdicts are possible: Pass: If the test case passed Fail: If the test case failed Inc: If the test case is inconclusive.
Observations:	Provides a reference to additional information relevant to the test presented in section 6.

Test Case Id	Role	Applicable	Run	Verdict	Observations

B.2.3 Observations

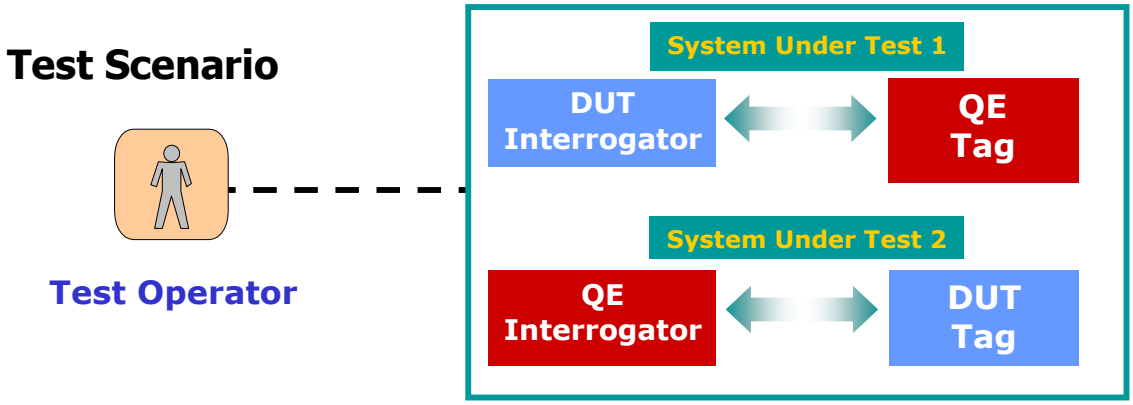
ANNEX C (Informative) Qualified Equipment

C.1 Qualified Equipment Purpose

In order to ensure that interoperability tests are repeatable and consistent, a set of products should be identified for use during interoperability testing – these products will be called Qualified Equipment (QE). QEs are standard commercial products that have been selected for the purpose of interoperability testing and shown to be fully compliant with reference standards.

QE (Qualified Equipment): Grouping of one or more devices that has been shown, by rigorous and well-defined testing, to interoperate with other equipment

Figure C-1 Interoperability Test Scenario



C.2 Selection of QE

1. GS1 EPCglobal sends a Request to invite manufacturers to submit products or prototypes of both tags and interrogators to become QE.
2. A selection process takes place. The selection process involves:
   a. Evaluation of the characteristics of the products, including supported features.
   b. Testing of interoperability within the set of submitted equipment.
   c. Evaluation and Performance: Predictability of results and overall performance.
   d. Maximum representation of technologies and manufacturing process.
   e. Evaluation of usability and features to optimize testing such as short read times and consistent measurement distances.
   f. Life-time of product and maintenance.
3. The Test Labs will produce a report providing guidance and a recommendation for selection of two or more QE of both tags and interrogators.
4. The Test Labs will provide a list of potential QEs to GS1 EPCglobal for ratification of the selection. The Test Labs nominates the equipment as Preliminary Qualified Equipment (Pre-QE).
5. Pre-QE is used in the Hardware Certification Testing Program to run interoperability testing.
6. If no problems are detected during operation and products successfully compliance EPCglobal Gen-2 Conformance Testing, Pre-QE officially becomes QE.
7. As the standard develops (changes, errata process, new features, etc.) the process to select new QE is performed again.

C.3 Evaluation of the products

For Interrogators:

- The QE shall operate over all the declared frequency plans and sub-bands specified in the EPCglobal Gen-2 Protocol.
- The QE shall support the maximum number of combinations of modulation and codification formats, Tari/RTCal/TRCal values, etc.
- The QE shall implement a suitable combination of optional commands and optional features.
- The QE will employ an intuitive GUI that allows test operators to run Test Cases in customizable suites.

For Tags:

- The QE shall implement a suitable combination of optional commands, optional features and optional memory banks and/or passwords to cover the most significant implementations.