All that Identification

ATID Co.,Ltd

RFID API Reference Guide for Android Developers

Android Developer Guide

SW Team

2022-06-20

Revision History

Ver.	Date	Reason[1]	Description[2]	Author
v 1.0	2021-07-14	Draft	Initial draft	SW Team
v 1.1	2022-06-20	add	Added ISO 18000-6B and Rail tag APIs	SW Team

[1] Revision : Define the contents are addition/modification/deletion

[2] Description: Describe revised page number and contents

Contents

1. Intro

This document is aim to describe usage of SDK Library for Android developers, to develop an application by using RFID SDK Library.
Development tool contained in this document, Android Studio, supports Android 10.

Dependency libraries description,

Library	Description
atid.dev.rfid	Library for Android, to control RFID Module
atid.system.comm	Library for Android, to control communication with Module
atid.system.jcomm	Library for Android, to control communication with Module
atid.system.ctrl	Library for Android, to control power of Module
atid.system.device	Library for Android, to manage information of Module
atid.util	Utility Library for internal use of SDK Library

2. Reference Library Guide

2.1. ATRfidManager Class

ATRfidManager is to control the creation and resource of RFID's Instance, which is installed in PDA. Also, it is Class that manages RFID resources between activities.

2.1.1. Method

2.1.1.1. getInstance

Creates RFID reader Object and connects RFID Module with RFID Object.

- **Syntax**

```
public static ATRfidReader getInstance()
```

- **Return value**

Returns Instance of RFID reader.

- **Remarks**

Once getInstance method works successfully, it creates and returns Instance of Reader.
Calls from onCreat method of Main Activity.

2.1.1.2. onDestroy

Finishes RFID Reader's connection with RFID Module which is created from getInstance, then releases RFID Reader's object.

- **Syntax**

```
public static void onDestroy()
```

- **Remarks**

onDestroy method is to set RFID reader object's resources free, then execute the releasing process.
Calls from Main Activity's on Destroy method.

2.1.1.3. wakeUp

Calls to wakeup RFID Module, in sleep mode.

- **Syntax**

```
public static void wakeUp()
```

- **Remarks**

Calls from onStart method of App's all Activity. Calls from sleep method. RFID Module is not working if wakeup method has been called.
Calling WakeUp and sleep must be in pairs.

2.1.1.4. sleep

Calls RFID Module to sleep in Wakeup mode.

- **Syntax**

```
public static void sleep()
```

- **Remarks**

Calls from onStop method of Application's all Activity. If sleep is not called in onStop method, Module operates continuously even entered Sleep mode by pressing power button of PDA.

Calling of wakeUp and sleep must be in pairs.

2.1.1.5. getVersion

Returns version of Library.

- **Syntax**

```
public static String getVersion()
```

- **Remarks**

Returns version of atid.dev.rfid.jar in use.

2.1.1.6. checkAutoModule

Returns Instance by searching RFID Module installed in PDA.

- **Syntax**

```
public static ATRfidReader checkAutoModule ()
```

- **Remarks**

Returns Instance of RFID Reader according to all RFID Module searched, which supports the PDA.

Library is internal use only, not for temporary use.

2.1.1.7. checkModule

Returns Instance according to provided RfidModuleType.

- **Syntax**

```
public static ATRfidReader checkModule(RfidModuleType type)
```

- **Parameters**

type: Module Type of Instance to create.

- **Remarks**

Creates Instance of inputted Module Type.

Library is internal use only, not for temporary use.

2.2. ATRfidReader Class

ATRfidReader Class creates Instance of RFID Reader, then sets Instance of RFID Reading and Configuration.

2.2.1. Method

2.2.1.1. Reset

Resets RFID module.

- **Syntax**

```
public void Reset()
```

- **Remarks**

Recommends to use in only special cases, not for temporary use.

2.2.1.2. setLogLevel

Sets log level of message, that outputs to LogCat.

- **Syntax**

```
public void setLogLevel(int level)
```

- **Parameters**

level: message log level.

- **Remarks**

Only for library debugging use, not for temporary use.

2.2.1.3. destroy

destroy method is to destroy Instance of ATRfidReader by force.

- **Syntax**

```
public void destroy()
```

- **Remarks**

It generally is called from ATRfidManager, not necessary to call individually.

2.2.1.4. powerControl

powerControl method is to control RFID Module power.

- **Syntax**

```
public void powerControl(boolean enabled)
```

- **Parameters**

enabled: Turn on RFID Module, if true. Turn off RFID Module, if false.

- **Remarks**

Power of RFID Module controls by itself in ATRfidManager. Recommends to use in only special cases, not for temporary use.

2.2.1.5. connect

connect method is to execute connection with RFID Module.

- **Syntax**

```
public boolean connect()
```

- **Return value**

Returns true in proper connection, false if not.

- **Remarks**

Execute connect only once when connecting with RFID Device.

2.2.1.6. disconnect

disconnect method is to unlock the connection with RFID Module.

- **Syntax**

```
public void disconnect()
```

- **Remarks**

Execute disconnect only once when unlocking with RFID Device.

2.2.1.7. readEpc6cTag

readEpc6cTag method is to save ISO 1800-6C tag in Inventory as a single mode.

- **Syntax**

```
public ResultCode readEpc6cTag()
```

- **Return value**

Returns the result of command execution in ResultCode enumeration type.

- **Remarks**

readEpc6cTag method is to execute Inventory function according to ISO18000-6C Tag from RFID Module, if tag is read properly, tag data gets transmitted through onReaderReadTag method of RfidReaderEventListener.

2.2.1.8. readEpc6bTag

readEpc6cTag method is to save ISO 18000-6B tag in Inventory as a single mode.

- **Syntax**

```
public ResultCode readEpc6cTag()
```

- **Return value**

Returns result of command execution in ResultCode enumeration type.

- **Remarks**

readEpc6bTag method is to execute Inventory function according to ISO18000-6B Tag from RFID Module, if tag is read properly, tag data gets transmitted through onReaderReadTag method of RfidReaderEventListener.

2.2.1.9. readEpcRailTag

readEpc6cTag method is to save AEI/Rail tag in Inventory as a single mode.

- **Syntax**

```
public  ResultCode  readEpcRailTag()
```

- **Return value**

Returns the result of command execution in ResultCode enumeration type.

- **Remarks**

readEpcRailTag method is to execute Inventory function according to AEI/Rail Tag from RFID Module, if tag is read properly, tag data gets transmitted through onReaderReadTag method of RfidReaderEventListener.

2.2.1.10. inventory6cTag

inventory6cTag method is to save ISO 18000-6C tag in Inventory as a multiple mode.

- **Syntax**

```
public ResultCode inventory6cTag()
```

- **Return value**

Returns result of command execution in ResultCode enumeration mode.

- **Remarks**

Inventory6cTag method is execute Inventory function according to ISO18000-6C tag from RFID Module, if tag is read properly, tag data gets transmitted through onReaderReadTag method of RfidReaderEventListener.

2.2.1.11. inventory6bTag

inventory6cTag method is to save ISO 18000-6B tag in Inventory as a multiple mode.

- **Syntax**

```
public ResultCode inventory6bTag()
```

- **Return value**

Returns result of command execution in ResultCode enumeration mode.

- **Remarks**
- Inventory6cTag method is execute Inventory function according to ISO18000-6B tag from RFID Module, if tag is read properly, tag data gets transmitted through onReaderReadTag method of RfidReaderEventListener.

2.2.1.12. inventoryRailTag

inventory6cTag method is to save ISO 18000-6C tag in Inventory as a multiple mode.

- **Syntax**

```
public ResultCode inventoryRailTag()
```

- **Return value**

Returns result of command execution in ResultCode enumeration mode.

- **Remarks**

Inventory6cTag method is execute Inventory function according to AEI/Rail tag from RFID Module, if tag is read properly, tag data gets transmitted through onReaderReadTag method of RfidReaderEventListener.

2.2.1.13. readMemory6c

readMemory6c method is to execute Read Memory function according to ISO 18000-6C tag.

- **Syntax**

```
public ResultCode readMemory6c(BankType bank, int offset, int length)
public ResultCode readMemory6c(BankType bank, int offset, int length,
       String password)
public ResultCode readMemory6c(BankType bank, int offset, int length,
       EpcMatchParam epc)
public ResultCode readMemory6c(BankType bank, int offset, int length,
       String password , EpcMatchParam epc)
```

- **Parameters**

bank : appoints memory bank of tag.

offset : appoints initial address of data in word unit.

length : appoints length of data in word unit.

password : appoints access password in 4Byte Hex word.

epc : appoins EPC information of tag, when executing read memory according to tag.

- **Return value**

Returns result of command execution in ResultCode enumeration type.

- **Remarks**

readMemory6c method is to execute read memory function according to ISO18000-6C tag from RFID module, if tag is read properly, result get transmitted through onReaderResult method of RfidReaderEventListener

2.2.1.14. readMemory6b

readMemory6b method is to execute Read Memory function according to ISO 18000-6B tag.

➢ **Syntax**

```
public ResultCode readMemory6c(int offset, int length)
```

➢ **Parameters**

offset : Appoints starting address of tag data in unit of byte.

length : Appoints length of tag data in unit of byte.

➢ **Return value**

Returns result of command execution in ResultCode enumeration mode.

➢ **Remarks**

readMemory6b method is to execute read memory function according to ISO18000-6B tag from RFID module, if tag is read properly, result get transmitted through onReaderResult method of RfidReaderEventListener

This method only works when return value of getModuleType is I900MA.

2.2.1.15. writeMemory6c

readMemory6c method is to execute Write Memory function according to ISO 18000-6C tag.

➢ **Syntax**

```
public ResultCode writeMemory6c(BankType bank, int offset, String data)
public ResultCode writeMemory6c(BankType bank, int offset, String data,
        String password)
public ResultCode writeMemory6c(BankType bank, int offset, String data,
        EpcMatchParam epc)
public ResultCode writeMemory6c(BankType bank, int offset, String data,
        String password, EpcMatchParam epc)
```

➢ **Parameters**

bank : appoints memory bank of tag.

offset : appoints initial address of data in word unit.

length : appoints length of data in word unit.

password : appoints access password in 4Byte Hex word.

epc : appoins EPC information of tag, when executing read memory according to tag.

➢ **Return value**

Returns result of command execution in ResultCode enumeration mode.

➢ **Remarks**

writeMemory6C method is to execute write memory function according to ISO18000-6C tag from RFID module, if tag is written properly, result get transmitted through onReaderResult method of RfidReaderEventListener.

2.2.1.16. writeMemory6b

readMemory6b method is to execute Write Memory function according to ISO 18000-6B tag.

➢ **Syntax**

```
public ResultCode writeMemory6b(int offset, String data)
```

➢ **Parameters**

offset : appoints initial address of data in word unit.

data : appoints data to write in unit of byte by Hex word

➢ **Return value**

Returns result of command execution in ResultCode enumeration mode.

➢ **Remarks**

➢ writeMemory6b method is to execute write memory function according to ISO18000-6B tag from RFID module, if tag is written properly, result get transmitted through onReaderResult method of RfidReaderEventListener.

This method only works when return value of getModuleType is I900MA

2.2.1.17. lock6c

lock6c method is to execute Lock function according to ISO 18000-6C tag

➢ **Syntax**

```
public ResultCode lock6c(LockParam param)
public ResultCode lock6c(LockParam param, String password)
public ResultCode lock6c(LockParam param, EpcMatchParam epc)
public ResultCode lock6c(LockParam param, String password, EpcMatchParam epc)
```

➢ **Parameters**

param : Instance of LckParam including information according to memory bank to execute lock

password : Appoints Access Password of tag in 4Byte Hex word.

epc : Appoints epc information of tag when executing lock for specific tag

➢ **Return value**

Returns result of command execution in enumeration type.

➢ **Remarks**

Lock6c method is to execute lock function according to ISO18000-6C tag from RFID Module, it tag is locked properly, result get transmitted through onReaderResult of RfidReaderEventListener.

2.2.1.18. kill6c

killk6c method is to execute kill function according to ISO 18000-6C tag

➢ **Syntax**

```
public ResultCode kill6c(String password)
public ResultCode kill6c(String password, EpcMatchParam epc)
```

➢ **Parameters**

password : Appoints Access Password of tag in 4Byte Hex word.

epc : Appoints epc information of tag when executing lock for specific tag

➢ **Return value**

Returns result of command execution in ResultCode enumeration type.

➢ **Remarks**

kill6c method is to execute kill function according to ISO18000-6C tag from RFID Module, it tag is killed properly, result get transmitted through onReaderResult of RfidReaderEventListener.

2.2.1.19. stop

Stop method is to sop all actions including Inventory, Read / Write Memory, Lock, Kill.

➢ **Syntax**

```
public ResultCode stop()
```

➢ **Return value**

Returns result of command execution in ResultCode enumeration type.

➢ **Remarks**

Stops all actions and change status to pause.

2.2.1.20. saveProperties

Saves attribute value of changed RFID Module.

➢ **Syntax**

```
public ResultCode saveProperties()
```

➢ **Return value**

Returns result of command execution in ResultCode enumeration type.

➢ **Remarks**

Saves attribute value of changed RFID Module, maintains former value after restart of application

This method only works when return value of getModuleType is I900MA

2.2.1.21. defaultProperties

defaultProperties method is to execute initialization of all attribute values of RFID Module.

➢ **Syntax**

```
public ResultCode defaultProperties()
```

➢ **Return value**

Returns result of command execution in ResultCode enumeration type.

➢ **Remarks**

defaultProperties method initializes all setted attribute values of RFID Module.

2.2.1.22. getFirmewareVersion

Returns firmware version of RFID Module.

➢ **Syntax**

```
public String getFirmwareVersion() throws ATRfidReaderException
```

➢ **Return value**

Returns firmware version in word.

➢ **Remarks**

getFirmewareVersion method must be executed after calling getInstance.

2.2.1.23. getState

getState method returns the connection status between Reader Object and RFID Module.

➢ **Syntax**

```
public ConnectionState getState()
```

➢ **Return value**

Returns connection status, please refer to ConnectionState for description.

➢ **Remarks**

getState method must be used after calling getInstance method.

2.2.1.24. getAction

getState method returns action status of RFID Module.

➢ **Syntax**

```
public ActionState getAction()
```

➢ **Return value**

Returns action status of RFID Module, please refer to ActionState for description.

➢ **Remarks**

getAction method must be used after calling getInstance method.

2.2.1.25. getOperationTime

Returns operation time of RFID module.

- **Syntax**

```
public int getOperationTime() throws ATRfidReaderException
```

- **Return value**

Returns operating time in ms unit.

- **Remarks**

Returns operating time of Module in integer type, in ms unit. If 0 is set for value, it stops Module or operates until the completion of order in operation.

2.2.1.26. setOperationTime

setOperationTime method sets operation time of RFID Module.

- **Syntax**

```
public void setOperationTime(int time) throws ATRfidReaderException
```

- **Parameters**

time : Operation time of Module in ms unit.

- **Remarks**

Sets operating time of Module in integer type, in ms unit. If 0 is set for value, it stops Module or operates until the completion of order in operation.

2.2.1.27. getPowerRange

Returns minimum and maximum power level of Antenna.

- **Syntax**

```
public RangeValue getPowerRange() throws ATRfidReaderException
```

- **Return value**

Returns Instance of RangeValue Class, according to minimum and maximum power level as per country set value of RFID Module.

- **Remarks**

Changes on country set status of RFId Module.

2.2.1.28. getPower

getPower method returns power level of Antenna

- **Syntax**

```
public int getPower() throws ATRfidReaderException
```

- **Return value**

Returns power level of Antenna in integer type of *10.

➢ **Remarks**

Outputs *10 of returned level. If output of Antenna is set 30dbm, 300 will be returned, which is *10 of 30dbm. Returned level is between minimum and maximum level evaluated from getPowerRange method.

2.2.1.29. setPower

setPower method sets Antenna output level of RFID Module.

➢ **Syntax**

```
public void setPower(int power) throws ATRfidReaderException
```

➢ **Parameters**

power : Antenna output level * 10 in integer type.

➢ **Remarks**

Sets *10 of Antenna output level. If output level to set is 30dbm, set 300. Set level is between minimum and maximum level evaluated from getPowerRange method.

2.2.1.30. getAntennaCycleCount

getAntennsCycleCount method returns Antenna Cycle count which is needed for operation execution.

➢ **Syntax**

```
public int getAntennaCycleCount() throws ATRfidReaderException
```

➢ **Return value**

Returns Antenna Cycle count in integer type.

➢ **Remarks**

Set value is 0~65535, effects overall performance of Module. It is not to be used for temporary.

2.2.1.31. setAntennaCycleCount

setAntennaCycleCount method returns Antenna Cycle count which is needed for operation execution.

➢ **Syntax**

```
public void setAntennaCycleCount(int count) throws ATRfidReaderException
```

➢ **Parameters**

count : Antenna Cycle count

➢ **Remarks**

Set value is 0~65535, effects overall performance of Module. It is not to be used for temporary.

2.2.1.32. getDWellTime

getDWellTime method returns dwell time of antenna.

➢ **Syntax**

```
public int getDWellTime() throws ATRfidReaderException
```

➢ **Return value**

Returns dwell time in integer type.

➢ **Remarks**

It effects overall performance of Module. It is not to be used for temporary.

2.2.1.33. setDWellTime

setDwellTime method sets dwell time of antenna.

➢ **Syntax**

```
public void setDWellTime(int time) throws ATRfidReaderException
```

➢ **Parameters**

time : dwell time

➢ **Remarks**

It effects overall performance of Module. It is not to be used for temporary.

2.2.1.34. getInventoryRoundCount

getInventoryRoundCount method returns inventory round count.

➢ **Syntax**

```
public int getDWellTime() throws ATRfidReaderException
```

➢ **Return value**

Returns dwell time in integer type.

➢ **Remarks**

➢ It effects overall performance of Module. It is not to be used for temporary.

2.2.1.35. setInventoryRoundCount

wetDWellTime method sets inventory round count.

➢ **Syntax**

```
public void setDWellTime(int count) throws ATRfidReaderException
```

➢ **Parameters**

time : inventory round count

➢ **Remarks**

- It effects overall performance of Module. It is not to be used for temporary.

2.2.1.36. getInventoryTime

getInventoryTime method returns time of Antenna in activation from Inventory Round time of RFID Module.

- **Syntax**

```
public int getInventoryTime() throws ATRfidReaderException
```

- **Return value**

Returns time of Antenna in activation, in ms unit.

- **Remarks**

RFID Module carries Inventory Time and Idle Time during one Inventory Round Time. Maximum Inventory Round Time is 400ms, and sum of Inventory Time and Idle Time can't exceed it.

2.2.1.37. setInventoryTime

getInventoryTime method sets time of Antenna in activation from Inventory Round time of RFID Module.

- **Syntax**

```
public void setInventoryTime(int time) throws ATRfidReaderException
```

- **Parameters**

time : Time of Antenna in activation, in ms unit.

- **Remarks**

Please refer to getInventoryTime.

2.2.1.38. getIdleTime

getIdleTime method returns valid time of Antenna from Inventory Round of RFID Module.

- **Syntax**

```
public int getIdleTime() throws ATRfidReaderException
```

- **Return value**

Returns valid time of Antenna, in ms unit.

- **Remarks**

RFID Module carries Inventory Time and Idle Time during one Inventory Round Time. Maximum Inventory Round Time is 400ms, and sum of Inventory Time and Idle Time can't exceed it.

2.2.1.39. setIdleTime

setIdleTime method sets valid time of Antenna from Inventory Round Time of RFID Module.

➢ **Syntax**

```
public void setIdleTime(int time) throws ATRfidReaderException
```

➢ **Parameters**

time : Time of Antenna in activation, in ms unit.

➢ **Remarks**

Please refer to getIdleTime.

2.2.1.40. getReportRssi

getReportRssi method returns whether to report EPC value from executing Inventory of RIFD Module with RSSI value, or not with RSSI value.

➢ **Syntax**

```
public boolean getReportRssi() throws ATRfidReaderException
```

➢ **Return value**

Boolean type after deciding whether with RSSI value or, not with RSSI value.

➢ **Remarks**

If returned value is true, it includes RSSI value in data, for ReaderReadTag event.

2.2.1.41. setReportRssi

setReportRssi method sets whether to report EPC value from executing Inventory of RIFD Module with RSSI value, or not with RSSI value

➢ **Syntax**

```
public void setReportRssi(boolean enabled) throws ATRfidReaderException
```

➢ **Parameters**

enabled : Boolean type after deciding whether with RSSI value or, not with RSSI value.

Remarks

If returned value is true, it includes RSSI value in data, for ReaderReadTag event.

2.2.1.42. getInventorySession

getInventorySession method returns Tag Session from Inventory execution of RFID Module.

➢ **Syntax**

```
public InventorySession getInventorySession() throws ATRfidReaderException
```

➢ **Return value**

Tag session in InventorySession type.

2.2.1.43. setInventorySession

setInventorySession method sets Tag Session from Inventory execution of RFID Module.

➢ **Syntax**

```
public void setInventorySession(InventorySession session) throws
ATRfidReaderException
```

➢ **Parameters**

session : InventorySession which show Tag Session, in enumeration type.

2.2.1.44. getInventoryTarget

gettInventoryTarget method returns Tag Session status from Inventory execution of RFID Module.

➢ **Syntax**

```
public InventoryTarget getInventoryTarget() throws ATRfidReaderException
```

➢ **Return value**

InventoryTarget which show Tag Session, in enumeration type.

2.2.1.45. setInventoryTarget

settInventorySession method sets Tag Session status from Inventory execution of RFID Module.

➢ **Syntax**

```
public void setInventoryTarget(InventoryTarget target) throws
ATRfidReaderException
```

➢ **Parameters**

target : InventoryTarget which show Tag Session, in enumeration type.

2.2.1.46. getSelectFlag

getSelectFlag method sets Tag SL status from Inventory execution.

➢ **Syntax**

```
public SelectFlagType getSelectFlag() throws ATRfidReaderException
```

➢ **Return value**

SelectFlagType which shows Tag SL status in enumeration type.

2.2.1.47. setSelectFlag

setSelectFlag method sets Tag SL status from Inventory execution.

➢ **Syntax**

```
public void setSelectFlag(SelectFlagType type) throws ATRfidReaderException
```

- **Parameters**

  type : SelectFlagType which shows Tag SL status in enumeration type.

2.2.1.48. getUseSelectionMask

getUseSelectionMask method returns whether to use Selection Mask, or not, when executing Inventory of RFID Module or other orders.

- **Syntax**

```
public boolean getUseSelectionMask() throws ATRfidReaderException
```

- **Return value**

  Boolean type after deciding whether to use Selection Mask or not.

- **Remarks**

  If returned value is true, it can execute Inventory or other orders by using Selection Mask.

2.2.1.49. setUseSelectionMask

setUseSelectionMask method sets whether to use Selection Mask, or not, when executing Inventory of RFID Module or other orders.

- **Syntax**

```
public void setUseSelectionMask(boolean used) throws ATRfidReaderException
```

- **Parameters**

  enabled : Boolean type after deciding whether to use Selection Mask or not.

- **Remarks**

  If returned value is true, it can execute Inventory or other orders by using Selection Mask.

2.2.1.50. getSelectionMask6c

getSelectionMask6c method returns value of Selection Mask set.

- **Syntax**

```
public SelectionMask6c getSelectionMask6c(int index) throws ATRfidReaderException
```

- **Parameters**

  index : Index (0~7) of Selection Mask sequence to get returned.

- **Return value**

  Instance of SelctionMask6c Class which has information of Selection Mask set in appointed index.

- **Remarks**

  RFID Module can set Selection Mask up to 8.

2.2.1.51. setSelectionMask6c

getSelectionMask6c method sets value of Selection Mask set.

- **Syntax**

```
public void setSelectionMask6c(int index, SelectionMask6c mask) throws
ATRfidReaderException
```

- **Parameters**

index : Index (0~7) of Selection Mask sequence to get returned.

mask : Instance of SelectionMask6c Class which has information of Selection Mask to set.

Remarks

RFID Module can set Selection Mask up to 8.

2.2.1.52. getSelectionMask6cList

getSelectionMask6cList method returns list of Selection Mask values set in RFID Module.

- **Syntax**

```
public SelectionMask6cList getSelectionMask6cList() throws ATRfidReaderException
```

- **Return value**

Instance of SelectionMask6cList Class which has information of Selection Mask set.

- **Remarks**

RFID Module can set Selection Mask up to 8.

2.2.1.53. setSelectionMask6cList

setSelectionMask6cList method sets list of Selection Mask values in RFID Module.

- **Syntax**

```
public void setSelectionMask6cList(SelectionMask6cList masks) throws
ATRfidReaderException
```

- **Parameters**

masks : Instance of SelectionMask6cList Class which has information of Selection Mask to set.

- **Remarks**
- RFID Module can set Selection Mask up to 8.

2.2.1.54. getGlobalBand

getGlobalBand method returns country frequency fragrance quality of RFID Module.

- **Syntax**

```
public GlobalBandType getGlobalBand() throws ATRfidReaderException
```

- **Return value**

GlobalBandType which shows frequency fragrance quality in enumeration type.

➤ **Remarks**

Returns country information which shows frequency fragrance quality.

2.2.1.55. getFreqChannelCount

getFreqChannelCount method returns maximum number of frequency channel, in frequency channel table of RFID Module.

➤ **Syntax**

```
public int getFreqChannelCount() throws ATRfidReaderException
```

➤ **Return value**

Shows maximum number of frequency channel table in integer type.

➤ **Remarks**

RFID Module supports multiple number of frequency channel as per country frequency fragrance quality.

2.2.1.56. isUseFreqChannel

isUseFreqChannel method returns the status whether in using relevant index of channel, or not, among frequency channels.

➤ **Syntax**

```
public boolean isUseFreqChannel(int index) throws ATRfidReaderException
```

➤ **Parameters**

index : Shows index of frequency channel table to status whether in using channel, or not, get returned, in integer type

➤ **Return value**

Boolean type after deciding whether to use frequency channel or not.

➤ **Remarks**

Returns the status whether in using relevant index of channel, or not, among frequency channels. Index must be more than 0, and less than value which is returned to getFreqChannelCount method.

2.2.1.57. setUseFreqChannel

setUseFreqChannel method sets the status whether in using relevant index of channel, or not, among frequency channels.

➤ **Syntax**

```
public void setUseFreqChannel(int index, boolean isUsed) throws
ATRfidReaderException
```

➤ **Parameters**

index : Shows index of frequency channel table to status whether in using channel, or not, get returned, in integer type .

isUsed : Boolean type after deciding whether to use frequency channel or not.

- **Remarks**

Sets the status whether in using relevant index of channel, or not, among frequency channels. Index must be more than 0, and less than value which is returned to getFreqChannelCount method.

2.2.1.58. getChannelFreq

getChannelFreq method returns frequency value of relevant index among frequency channel.

- **Syntax**

```
public int getChannelFreq(int index) throws ATRfidReaderException
```

- **Parameters**

index : Shows index of frequency channel table to status whether in using channel, or not, get returned, in integer type .

- **Return value**

Shows frequency value in integer type.

- **Remarks**

Returns frequency value of relevant index among frequency channels. Index must be more than 0, and less than value which is returned to getFreqChannelCount method.

2.2.1.59. setEventListener

Sets to enable event from application.

- **Syntax**

```
public void setEventListener(RfidReaderEventListener listener)
```

- **Parameters**

listener : Interface created to handle special events(RFID) from application.

- **Remarks**

setEventListener method must be used after calling getInstance.

2.2.1.60. removeEventListener

Sets to disenable event from application.

- **Syntax**

```
public void removeEventListener(RfidReaderEventListener listener)
public void removeEventListener()
```

- **Parameters**

listener : Interface created to handle special events(RFID) from application.

2.3. ATRfidATX00S1Reader Class

ATRfidATX00S1Reader Class is Class from ATRfidReader, and additionally provides subordinative fucntions in R2000 Module.

To use this Class, Type Cast ATRfidReader Instance which is returned to getInstance method of ATFfidManager, to ATRfidATX00S1Reader.

2.3.1. Method

2.3.1.1. readMemory6cEx

Maintain CW as appointed time, and execute Read Memory function once or consecutively.

➢ **Syntax**

```
public ResultCode readMemory6cEx(boolean isContinuousMode, BankType bank, int
offset, int length, String password, EpcMatchParam epc, int delay);
```

➢ **Parameters**

isContinuousMode : Appoints whether to execute consecutively or not.

bank : Appoints Memroy Bank of Tag.

offset : Appoints starting address of Tag Data in word unit.

length : Appoints length of Tag Data in WORD unit.

password : Appoints Access Password of Tag in 4Byte Hex word.

epc : Appoints EPC data of Tag, when executing Read Memory for a certain Tag.

delay : Appoints EPC data of Tag, when executing Read Memory for a certain Tag.

➢ **Return value**

Returns result of command execution in ResultCode enumeration type.

➢ **Remarks**

readMemory6c method is to execute Read Memory function for ISO18000-6C Tag in RFID Module, and result is convyed through onReaderResult Method of RfidReaderEventListener when executed properly.

2.3.1.2. readMemory6cSync

Executes same function as readMemory6cEx, but returns data of tags read in result of Method execution.

➢ **Syntax**

```
public String readMemory6cSync(boolean isContinuousMode, BankType bank, int
offset, int length, SelectionMask6cList masks, String password, EpcMatchParam
epc, int delay);
```

➢ **Parameters**

isContinuousMode : Appoints whether to execute consecutively or not.

bank : Appoints Memory Bank of Tag.

offset : Appoints starting address of Tag Data in word unit.

length : Appoints length of Tag Data in WORD unit.

masks : Selection mask list of Tag to read.

password : Appoints Access Password of Tag in 4Byte Hex word.

epc : Appoints EPC data of Tag, when executing Read Memory for a certain Tag.

delay : Appoints time to maintain CW in millisecond unit.

- **Return value**

Returns tag data when Read memory is read properly, and returns null in case of failure or error.

- **Remarks**

This method doesn't return until function terminates, users are recommended to use after setting execution time by setOperationTime method.

2.3.1.3. readOemData

Reads OEM Register value of R2000 Module.

- **Syntax**

```
public int readOemData(int address);
```

- **Parameters**

address : OEM Register address to access (32bits value)

- **Return value**

value of written address (32bits value)

- **Remarks**

Reads Register value of R2000 Module directly, not using for general use.

2.3.1.4. writeOemData

Writes OEM Register value of R2000 Module.

- **Syntax**

```
public boolean writeOemData(int address, int value);
```

- **Parameters**

address : OEM Register address to access (32bits value)

value : Value to write in address (32bits value)

- **Return value**

Shows whether Method execution is succeed or not in boolean type.

- **Remarks**

Writes Register value of R2000 Module directly, not using for general use.

2.3.1.5. isUseLinkProfile

Returns whether Link Profile has been used or not.

➢ **Syntax**

```
public boolean isUseLinkProfile();
```

➢ **Return value**

Shows whether Link Profile has been used or not in boolean type.

2.3.1.6. getLinkProfileCount

Returns number of Link Profile.

➢ **Syntax**

```
public int getLinkProfileCount() throws ATRfidReaderException
```

➢ **Return value**

Shows number of Link Profile in integer type.

2.3.1.7. getActiveLinkProfile

Returns Index of Link Profile in use.

➢ **Syntax**

```
public int getActiveLinkProfilet() throws ATRfidReaderException
```

➢ **Return value**

Shows Link Profile in integer type.

2.3.1.8. setActiveLinkProfile

Changes Index of Link Profile in use.

➢ **Syntax**

```
public boolean setActiveLinkProfilet(int index) throws ATRfidReaderException
```

➢ **Parameters**

index : Link Profile index to change.

➢ **Return value**

Shows whether Method execution is succeed or not in boolean type.

2.3.1.9. getUseDefaultLinkProfile

Returns whether Default Link Profile is in use or nor

➢ **Syntax**

```
public boolean getUseDefaultLinkProfilet(int index) throws ATRfidReaderException
```

➢ **Parameters**

index : Link Profile index

➢ **Return value**

If value is 1, use Default Link Profile, and 0, use default of F/W.

2.3.1.10. **setDefaultLinkProfile**

Sets Index of Default Link Profile.

➢ **Syntax**

```
public boolean setDefaultLinkProfilet(int index, int used) throws
ATRfidReaderException
```

➢ **Parameters**

index : Link Profile index to set as Default.

used : Decides set value of index whether to us Default or not. (1: use, 2: not use)

➢ **Return value**

Shows whether Method execution is succeed or not in boolean type.

2.3.1.11. **setCarrierWaveDelay**

Sets time to maintain CW.

➢ **Syntax**

```
public boolean setCarrierWaveDelay(int delay) throws ATRfidReaderException
```

➢ **Parameters**

delay : Duration time (0~255 millisecond unit)

➢ **Return value**

Shows whether Method execution is succeed or not in boolean type.

2.3.1.12. **getCurrentSingulationAlgorithm**

Returns inventory algorithm to use.(default: DYNAMICQ)

➢ **Syntax**

```
public SingulationAlgorithm getCurrentSingulationAlgorithm() throws
ATRfidReaderException
```

➢ **Return value**

Shows inventory algorithm in SingulationAlgorithm type.

2.3.1.13. **setCurrentSingulationAlgorithm**

Changes inventory algorithm to use

➢ **Syntax**

```
public void setCurrentSingulationAlgorithm(SingulationAlgorithm algorithm) throws
ATRfidReaderException
```

➢ **Parameters**

algorithm : inventory algorithm to change

➢ **Remarks**

Be careful because it affects performance.

2.3.1.14. getQValue

Returns Q value to use

➢ **Syntax**

```
public QValue getQValue() throws ATRfidReaderException
```

➢ **Return value**

Q value in QValue type.

2.3.1.15. setQValue

Changes Q value

➢ **Syntax**

```
public void setQValue(QValue q) throws ATRfidReaderException
```

➢ **Parameters**

q : Q value to change

➢ **Remarks**

Be careful because it affects performance.

When using the FIXEDQ, use start q as a fixed value.

When using the DYNAMICQ, use start q, min q, max q dynamically.

These values should be set as follows.

- Start q must be greater than or equal to min Q and less than or equal to max Q
- Min q must be less than or equal to start Q and max Q
- Max q must be greater than or equal to start Q and min Q

2.4. ATRfid900MAReader Class

ATRfid900MaReader Class is inherited from ATRfidReader, and provides subjectional functions in AMS Module additionally.

To use this Class, Type Cast ATRfidReader Instance which is returned to getInstance method of ATFfidManager, to ATRfid900MAReader.

2.4.1. Method

2.4.1.1. readEpcRailTag

Inventory Rail tag in Single mode.

- **Syntax**

```
public ResultCode readEpcRailTag();
```

- **Return value**

Returns command execution result in ResultCode enumeration type.

- **Remarks**

readEpcRailTag method executes Inventory function according to Rail Tag in RFID Module. If read properly, tag data is transmitted through onReaderReadTag Method of RfidReaderEventListener.

To use this Method, exclusive F/W for Rail Tag equipped Module must be used.

If returned version to getFirmwareVersion is ‘R’, it is exclusive F/W for Rail Tag, If ‘M’, it is standard F/W.

2.4.1.2. inventoryRailTag

Inventory Rail tag in Multiple mode

- **Syntax**

```
public ResultCode inventoryRailTag();
```

- **Return value**

Returns command execution result in ResultCode enumeration type.

- **Remarks**

readEpcRailTag method executes Inventory function according to Rail Tag in RFID Module. If read properly, tag data is transmitted through onReaderReadTag Method of RfidReaderEventListener.

To use this Method, exclusive F/W for Rail Tag equipped Module must be used.

If returned version to getFirmwareVersion is ‘R’, it is exclusive F/W for Rail Tag, If ‘M’, it is standard F/W.

2.4.1.3. readEpcAnyTag

Inventory without appointing tag type in single mode.

- **Syntax**

```
public ResultCode readEpcAnyTag();
```

- **Return value**

Returns command execution result in ResultCode enumeration type.

- **Remarks**

readEpcAnyTag method executes Inventory function according to ISO 18000-6B, ISO 18000-6C, Rail Tag in RFID Module. If read properly, tag data is transmitted through onReaderReadTag Method of RfidReaderEventListener.

If standard F/W is applied, this method works, but doesn't respond to Rail Tag (only responds to 6B, 6C tags)

If returned version to getFirmwareVersion is 'R', it is exclusive F/W for Rail Tag, If 'M', it is standard F/W.

2.4.1.4. inventoryAnyTag

Inventory without appointing tag type in multiple mode.

- **Syntax**

```
public ResultCode inventoryAnyTag();
```

- **Return value**

Returns command execution result in ResultCode enumeration type.

- **Remarks**

readEpcAnyTag method executes Inventory function according to ISO 18000-6B, ISO 18000-6C, Rail Tag in RFID Module. If read properly, tag data is transmitted through onReaderReadTag Method of RfidReaderEventListener.

If standard F/W is applied, this method works, but doesn't respond to Rail Tag (only responds to 6B, 6C tags)

If returned version to getFirmwareVersion is 'R', it is exclusive F/W for Rail Tag, If 'M', it is standard F/W.

2.4.1.5. readMemory6b

Executes ISO 18000-6B Read Memory function with selecting Tag.

- **Syntax**

```
public ResultCode readMemory6b(int offset, int length, SelectionMask6b mask);
```

- **Parameters**

offset : Appoints address of Tag Data to execute reading in byte unit.

length : Appoints length of Tag Data to execute reading in byte unit.

mask : Appoints memory information of Tag, which is to be executed.

- **Return value**

Returns command execution result in ResultCode enumeration type.

- **Remarks**

readMemory6b method executes Read Memory function according to ISO18000-6B Tag in RFID Module. If read properly, tag data is transmitted through onReaderReadTag Method of RfidReaderEventListener.

2.4.1.6. writeMemory6b

Executes ISO 18000-6B Write Memory function with selecting Tag.

- **Syntax**

```
public ResultCode writeMemory6b(int offset, String data, SelectionMask6b mask);
```

- **Parameters**

offset : Appoints address of Tag Data to execute reading in byte unit.

data : Appoints Data to record Tag in byte unit Hex word.

mask : Appoints memory information of Tag, which is to be executed.

- **Return value**

Returns command execution result in ResultCode enumeration type.

- **Remarks**

writeMemory6b method executes Write Memory function according to ISO18000-6B Tag in RFID Module. If written properly, tag data is transmitted through onReaderReadTag Method of RfidReaderEventListener.

2.5. RfidReaderEventListener Interface

2.5.1. Method

2.5.1.1. onReaderStateChanged

onReaderStateChange method returns connection status of RFID Module.

- **Syntax**

```
void onReaderStateChanged(ATRfidReader reader, ConnectionState state);
```

- **Parameters**

reader : Reader Object occurred event.

state : Shows connection status of RFID in ConnectionState enumeration type.

- **Remarks**

If connection status changes, call from Reader Object which is connected to RFID Module.

2.5.1.2. onReaderActionChanged

onReaderActionChange method returns action status of RFID Module.

- **Syntax**

```
void onReaderActionChanged(ATRfidReader reader, ActionState action);
```

- **Parameters**

reader : Reader Object occurred event.

state : Shows connection status of RFID in ActionState enumeration type.

- **Remarks**

If connection status changes, call from Reader Object which is connected to RFID Module.

2.5.1.3. onReaderReadTag

onReaderReadTag method returns EPC value of Tag read from readEpc6cTag method or inventory6c method.

- **Syntax**

```
void onReaderReadTag(ATRfidReader reader, String tag, float rssi, float phase);
```

- **Parameters**

reader : Reader Object occurred event.

tag : Shows EPC of Tag written in Inventory in Hex type word.

rssi : Shows RSSI value in float type.

phase : Shows Phase value in float type.

- **Remarks**

If RFID Module reads EPC data of Tag by Inventory function, calls from Reader Object.

2.5.1.4. **onReaderResult**

onReaderResult method returns Access command result such as Read Memory, Write Memory, or Kill.

➢ **Syntax**

```
void onReaderResult(ATRfidReader reader, ResultCode code, ActionState action,
String epc, String data, float rssi, float phase);
```

➢ **Parameters**

reader : Reader Object occurred event.

code : Shows Access command result in ResultCode enumeration type.

action : Shows Access command result in ActionState enumeration type.

epc : Shows EPC data of Access Tag in Hex word type.

data : If Access command is ReadMemory, shows Tag data in Hex word type.

rssi : Shows RSSI value in float type.

phase : Shows Phase value in float type.

➢ **Remarks**

If Access related command is executed, calls from Reader Object which is connected to RFID Module.

2.6. Parameter Classes

2.6.1. RangeValue Class

2.6.1.1. Constructor

Initialize new Instance of RangeValue, which shows the range.

➢ **Syntax**

```
public RangeValue()
public RangeValue(int min, int max)
```

➢ **Parameters**

min : Shows minimum value in integer.

max : Shows maximum value in integer.

➢ **Remarks**

Used for showing value of returned range from getPowerRange.

2.6.1.2. Property Methods

2.6.1.2.1. getMin

Returns minimum value of set range.

➢ **Syntax**

```
public int getMin()
```

➢ **Return value**

Shows minimum value set in Instance, in integer.

2.6.1.2.2. getMax

Returns maximum value of set range.

➢ **Syntax**

```
public int getMax()
```

➢ **Return value**

Shows maximum value set in Instance, in integer.

2.6.2. LockParam Class

2.6.2.1. Constructor

Initialize new Instance of LockParam, which shows the range.

➢ **Syntax**

```
public LockParam()
public LockParam(LockType killPassword, LockType accessPassword,
                                LockType epc, LockType tid, LockType user)
```

➢ **Parameters**

killPassword : Shows Lock action of Kill Password in LockType enumeration type.

accessPassword : Shows Lock action of Access Password in LockType enumeration type.

epc : Shows Lock action of EPC Memory Bank in LockType enumeration type.

tid : Shows Lock action of TID Memory Bank in LockType enumeration type.

user : Shows Lock action of User Memory Bank in LockType enumeration type.

- **Remarks**

Uses Parameter of lock6c method.

2.6.2.2. Property Methods

2.6.2.2.1. getKillPassword

Returns Lock action of Kill Password.

- **Syntax**

```
public LockType getKillPassword()
```

- **Return value**

Lock action of Kill Password in LockType enumeration type.

2.6.2.2.2. setKillPassword

Sets Lock action of Kill Password.

- **Syntax**

```
public void setKillPassword(LockType killPassword)
```

- **Parameters**

killPassword : Shows Lock action of Kill Password in LockType enumeration type.

2.6.2.2.3. getAccessPassword

Returns Lock action of Access Password.

- **Syntax**

```
public LockType getAccessPassword()
```

- **Return value**

Shows Lock action of Access Password in LcokType enumeration type.

2.6.2.2.4. setAccessPassword

Sets Lock action of Access Password.

- **Syntax**

```
public void setAccessPassword(LockType accessPassword)
```

- **Parameters**

accessPassword : Shows Lock action of Access Password in LockType enumeration type.

2.6.2.2.5. getEPC

Returns Lock action of EPC Memory Bank.

- **Syntax**

```
public LockType getEPC()
```

- **Return value**

Shows Lock action of EPC Memory Bank in LockType enumeration type.

2.6.2.2.6. setEPC

Sets Lock action of EPC Memory Bank.

- **Syntax**

```
public void setEPC(LockType epc)
```

- **Parameters**

epc : Shows Lock action of EPC Memory Bank in LockType enumeration type.

2.6.2.2.7. getTID

Returns Lock action of TID Memory Bank.

- **Syntax**

```
public LockType getTID()
```

- **Return value**

Shows Lock action of TID Memory Bank in LockType enumeration type.

2.6.2.2.8. setTID

Sets Lock action of TID Memory Bank.

- **Syntax**

```
public void setTID(LockType tid)
```

- **Parameters**

tid : Shows Lock action of TID Memory Bank in LockType enumeration type.

2.6.2.2.9. getUser

Returns Lock action of User Memory Bank.

- **Syntax**

```
public LockType getUser()
```

- **Return value**

Shows Lock action of User Memory Bank in LockType enumeration type.

2.6.2.2.10. setUser

Sets Lcok action of User Memory Bank.

➢ **Syntax**

```
public void setUser(LockType user)
```

➢ **Parameters**

user : Shows Lock action of User Memory Bank in LockType enumeration type.

2.6.3. SelectionMask6c Class

2.6.3.1. Constructor

Initialize new Instance of SelectionMask6c Class, which shows Selection Mask.

➢ **Syntax**

```
public SelectionMask6c()
public SelectionMask6c(MaskTargetType target, MaskActionType action,
                    BankType bank, int pointer, int length, String mask,
                    boolean truncate)
```

➢ **Parameters**

target : Shows Session of Tag in Mask target in MaskTargetType enumeration type.

action : Decides Session applies to Mask condition, in MaskActionType enumeration type.

bank : Shows Memory Bank of Tag in Mask condition target, in BankType enumeration type.

pointer : Starting address to compare Mask value from in integer. (bit unit)

length : Appoints length to compare Mask value in integer. (bit unit)

mask : Shows Mask value in Hex word type.

truncate : Shows how long to cut length of Mask value, in boolean type.

➢ **Remarks**

Uses getSelectionMask6c or setSelectionMask6c method.

2.6.3.2. Property Methods

2.6.3.2.1. isUsed

Returns whether to use the current set Selection Mask condition or not.

➢ **Syntax**

```
public boolean isUsed()
```

➢ **Return value**

Shows whether to use Selection Mask information or not, in boolean type.

2.6.3.2.2. setUsed

Sets whether to use the current set Selection Mask condition or not.

➢ **Syntax**

```
public void setUsed(boolean used)
```

- **Parameters**

  used : Shows whether to use Selection Mask information or not, in boolean type.

2.6.3.2.3. getTarget

Returns Session of Tag in Selection Mask target.

- **Syntax**

```
public MaskTargetType getTarget()
```

- **Return value**

  Shows Session of Tag in Mask target, in MaskTargetType enumeration type.

2.6.3.2.4. setTarget

Sets Session of Tag in Selection Mask target.

- **Syntax**

```
public void setTarget(MaskTargetType target)
```

- **Parameters**

  target : Shows Session of Tag in Mask target, in MaskTargetType enumeration type.

2.6.3.2.5. getAction

Returns how to set Session according to Selection Mask condition.

- **Syntax**

```
public MaskActionType getAction()
```

- **Return value**

  Decides Session setting according to Mask condition, in MaskActionType enumeration type.

2.6.3.2.6. setAction

Sets how to set Session according to Selection Mask condition.

- **Syntax**

```
public void setAction(MaskActionType action)
```

- **Parameters**

  action : Decides Session setting according to Mask condition, in MaskActionType enumeration type.

2.6.3.2.7. getBank

Returns Tag Memory Bank compared with Selection Mask.

- **Syntax**

```
public BankType getBank()
```

- **Return value**

  Shows Memory Bank of Tag in Mask condition target, in BankType enumeration type.

2.6.3.2.8. setBank

Sets Tag Memory Bank compared with Selection Mask.

- **Syntax**

```
public void setBank(BankType bank)
```

- **Parameters**

  bank : Shows Memory Bank of Tag in Mask condition target, in BankType enumeration type.

2.6.3.2.9. getPointer

Returns Starting address to compare with Mask value of Selection Mask.

- **Syntax**

```
public int getPointer()
```

- **Return value**

  Shows starting address to compare Mask value, in integer. (bit unit)

2.6.3.2.10. setPointer

Sets Starting address to compare with Mask value of Selection Mask.

- **Syntax**

```
public void setPointer(int pointer)
```

- **Parameters**
- **pointer :** Shows starting address to compare Mask value, in integer. (bit unit)

2.6.3.2.11. getLength

Returns length to compare with Mask value of Selection Mask.

- **Syntax**

```
public int getLength()
```

- **Return value**

  Decides length to compare Mask value in integer. (bit unit)

2.6.3.2.12. setLength

Sets length to compare with Mask value of Selection Mask.

- **Syntax**

```
public void setLength(int length)
```

- **Parameters**

- **length :** Decides length to compare Mask value in integer. (bit unit)

2.6.3.2.13. getMask

Returns Mask value to compare in Selection Mask.

- **Syntax**

```
public String getMask()
```

- **Return value**

Shows Mask value in Hex word type.

2.6.3.2.14. setMask

Sets Mask value to compare in Selection Mask.

- **Syntax**

```
public void setMask(String mask)
```

- **Parameters**

mask : Shows Mask value in Hex word type.

2.6.3.2.15. getTruncate

Returns whether to cut length as Mask value of Selection Mask or not.

- **Syntax**

```
public boolean getTruncate()
```

- **Return value**

Shows whether to cut length as Mask value in boolean type.

2.6.3.2.16. setTruncate

Sets whether to cut length as Mask value of Selection Mask or not.

- **Syntax**

```
public void setTruncate(boolean truncate)
```

- **Parameters**

truncate : Shows whether to cut length as Mask value in boolean type.

2.6.4. SelectionMask6b Class

2.6.4.1. Constructor

Initializes new Instance of SelectionMask6b Class, which shows Selection Mask.

- **Syntax**

```
public SelectionMask6b()
public SelectionMask6b(int pointer, String mask, MaskActionType action)
```

- **Parameters**

pointer : Starting address must be set in 0 to compare Mask value.

mask : Shows Mask value in Hex word type, UID(64bits) must be completely inputted.

action : Mask condition, must be set in MaskMatchingType.Match

- **Remarks**

Uses from readMemory6b, writeMemory6c method.

2.6.4.2. Property Methods

2.6.4.2.1. getPointer

Returns pointer value of Selection Mask currently set.

- **Syntax**

```
public int getPointer()
```

- **Return value**

Starting address to compare Mask value in integer.

2.6.4.2.2. setPointer

Sets Pointer value of Selection Mask.

- **Syntax**

```
public void setPointer(int pointer)
```

- **Parameters**

pointer : Shows starting address to compare Mask value in integer.

2.6.4.2.3. getAction

Returns Selection Mask condition.

- **Syntax**

```
public MaskActionType getAction()
```

- **Return value**

Decides Mask condition in MaskMatchingType in enumeration type.

2.6.4.2.4. setAction

Sets Selection Mask condition.

- **Syntax**

```
public void setAction(MaskActionType action)
```

- **Parameters**

action : Decides Mask condition in MaskMatchingType enumeration type.

2.6.4.2.5. getMask

Returns Mask value to compare with Selection Mask.

- **Syntax**

```
public String getMask()
```

- **Return value**

Shows Mask value in Hex word type.

2.6.4.2.6. setMask

Sets Mask value to compare with Selection Mask.

- **Syntax**

```
public void setMask(String mask)
```

- **Parameters**
- **mask :** Shows Mask value in Hex word type.

2.6.5. EpcMatchParam Class

2.6.5.1. Constructor

Initializes new Instance of EpcMatchParam.

- **Syntax**

```
public EpcMatchParam()
public EpcMatchParam(MaskMatchingType match, int offset, int length, String data)
```

- **Parameters**

match : Sets data whether to matching or non-matching with tag value.

offset : Offset to start data to compare with tag value. (EPC starting in 0, in bit unit)

length : Length of data to compare with tag value. (bit unit)

data : Data which will be compared with tag value.

- **Remarks**

Used for Access command such as Read/write/lock/kill.

2.6.5.2. Property Methods

2.6.5.2.1. getMatch

Returns set MaskMatchingType.

- **Syntax**

```
public MaskMatchingType getMatch()
```

- **Return value**

MaskMatchingType set in Instance.

2.6.5.2.2. setMatch

Sets MaskMatchingType.

- **Syntax**

```
public void setMatch(MaskMatchingType match)
```

- **Parameters**

match : MaskMatchingType

2.6.5.2.3. getOffset

Returns offset value of set data.

- **Syntax**

```
public int getOffset()
```

- **Return value**

Shows offset set in Instance, in integer. (bit unit)

2.6.5.2.4. setValue

Sets offset value of data.

- **Syntax**

```
public int setValue(int offset)
```

- **Parameters**

offset : Offset where data will be compared with tag value. (EPC starting in 0, in bit unit)

2.6.5.2.5. getLength

Returns length value of set data.

- **Syntax**

```
public int getOffset()
```

- **Return value**

Shows length set in Instance, in integer. (bit unit)

2.6.5.2.6. setLength

Sets length value of set data.

- **Syntax**

```
public void setLength(int length)
```

- **Parameters**

length : Length of data which will be compared with tag value. (bit unit)

2.6.5.2.7. getData

Returns set data of Instance.

- **Syntax**

```
public String getData()
```

- **Return value**

  Data which will be compared with tag value.

2.6.5.2.8. setData

Sets starting address to compare Mask value of Selection Mask.

- **Syntax**

```
public void setData(String data)
```

- **Parameters**

  data : Data which will be compared with tag value.

2.6.5.2.9. getValue

Returns set values in Instance after transferring to values which will be conveyed to RFID Module.

- **Syntax**

```
public int getValue()
```

- **Return value**

  Changed set value.

- **Remarks**

  It is an internal library being used, not for temporary use.

2.6.6. QValue Class

2.6.6.1. Constructor

Initializes new Instance of QValue.

- **Syntax**

```
public QValue()
public QValue(int start, int min, int max)
```

- **Parameters**

  start : starting q value

  min : minimum q value (SingulationAlgorithm=FIXED: N/A)

  max : maximum q value(SingulationAlgorithm=FIXED: N/A)

- **Remarks**

  Be careful because it affects performance.

2.6.6.2. **Property Methods**

2.6.6.2.1. getStartQ

Returns starting q value of instance

- **Syntax**

```
public int getStartQ()
```

- **Return value**

starting q value

- **Remarks**

When using the FIXEDQ, use start q as a fixed value.

When using the DYNAMICQ, start q must be greater than or equal to min q and less than or equal to max q.

2.6.6.2.2. setStartQ

Sets starting q value

- **Syntax**

```
public void setStartQ(int start)
```

- **Parameters**

start : starting q value to set(0~15)

- **Remarks**

When using the FIXEDQ, use start q as a fixed value.

When using the DYNAMICQ, start q must be greater than or equal to min q and less than or equal to max q.

2.6.6.2.3. getMinQ

Returns minimum q value of instance

- **Syntax**

```
public int getMinQ()
```

- **Return value**

minimum q value

- **Remarks**

When using DYNAMICQ, this value is valid.

Min q must be less than or equal to start q and max q.

2.6.6.2.4. setMinQ

Sets minimum q value

- **Syntax**

```
public void setMinQ(int min)
```

- **Parameters**

  start : minimum q value to set(0~15)

- **Remarks**

  When using DYNAMICQ, this value is valid.

  Min q must be less than or equal to start q and max q.

2.6.6.2.5. getMaxQ

Returns maximum q value of instance

- **Syntax**

```
public int getMaxQ()
```

- **Return value**

  maximum q value

- **Remarks**

  When using DYNAMICQ, this value is valid.

  Max q must be greater than or equal to start q and min q.

2.6.6.2.6. setMaxQ

Sets maximum q value

- **Syntax**

```
public void setMaxQ(int max)
```

- **Parameters**

  start : maximum q value to set(0~15)

- **Remarks**

  When using DYNAMICQ, this value is valid.

  Max q must be greater than or equal to start q and min q.

2.7. Enumerations

2.7.1. ActionState

The operating status of the RFID module.

Flag	Value	Description
Unknown	0x20	unknown
Inventory6bMulti	0x62	Multiple Inventory 6B Type
Inventory6bSingle	0x61	Single Inventory 6B Type
Inventory6cMulti	0x66	Multiple Inventory 6C Type
Inventory6cSingle	0x65	Single Inventory 6C Type
Inventory6cSelect	0x64	Select Inventory 6C Type
InventoryAnyMulti	0x6B	Multiple Inventory Any Type
ReadMemory6c	0x72	Read Memory 6C Type
ReadMemory6b	0x52	Read Memory 6B Type
WriteMemory6c	0x77	Write Memory 6C Type
WriteMemory6b	0x63	Write Memory 6B Type
Lock	0x6C	Lock Tag
Kill	0x6B	Kill Tag
Stop	0x33	Stop

2.7.2. BankType

Memory bank of the Tag which will be accessed

Flag	Value	Description
Reserved	0	Readerved Bank
EPC	1	EPC Bank
TID	2	TID Bank
User	3	User Bank

2.7.3. TagType

To set the Inventory tag type

Flag	Value	Description
Tag6C	0	ISO18000 6C
Tag6B	1	ISO18000 6B
TagRail	2	AEI/Rail
TagAny	3	N/A

2.7.4. ConnectionState

The connection status of Reader object

Flag	Value	Description
Disconnected	0	Connection lost
Connecting	2	Connection is in progress
Connected	3	Connection completed

2.7.5. InventorySession

Session which will be used while RFID module operating.

Flag	Value	Description
S0	0	Session 0
S1	1	Session 1
S2	2	Session 2
S3	3	Session 3

2.7.6. InventoryTarget

Target of Session which will be used while RFID module operating.

Flag	Value	Description
A	0	State A
B	1	State B
AB	2	A or B

2.7.7. LockType

The status of lock

Flag	Value	Description
NoChange	0	No effect
Unlock	1	Perform an unlock
Lock	2	Perform a lock
PermaLock	3	Perform a permalock

2.7.8. RfidModuleType

RFID module in use

Flag	Value	Description
None	0	No RFID module
I900MA	1	AMS Module (supported ISO 18000 6B/6C)

AT6EM_1	2	Not supported
AT9200P_1	3	Not supported
ATX00S_1	4	R2000 Module (supported ISO 18000 6C)

2.7.9. SelectFlagType

SL status of the Tag which will be used while RFID module operating.

Flag	Value	Description
All	0	assert or deassert
Deassert	1	deassert
Assert	2	assert

2.7.10. MaskMatchingType

The status of comparison of masking pattern and tag's data

Flag	Value	Description
Match	0x30	Matching
NonMatch	0x31	Non-Matching

2.7.11. GlobalBandType

Country setting for module

Flag	Value	Description
Korea	0	South Korea
Europe	1	Europe
NorthAmerica	2	North America
China	3	China
Taiwan	4	Taiwan
Brazil	5	Brazil
Malaysia	6	Malaysia
Hongkong	7	Hongkong
Japan1W	8	Japan (1W)
Japan250mW	9	Japan (250mW)
India	10	India
Indonesia	11	Indonesia
Japan125mW	12	Japan (125mW)
Israel	13	Israel
Australia	14	Australia
Newzealand	15	Newzealand
Philippines	16	Philippines
Singapore	17	Singapore

Thailand	18	Thailand
Uruguay	19	Uruguay
Vietnam	20	Vietnam
SouthAfrica	21	South Africa
Morocco	22	Morocco
Europe_B1	23	EN 302 208 Sub-bands b1
Europe_B2	24	EN 302 208 Sub-bands b2
Europe_B3	25	EN 302 208 Sub-bands b3
Peru	26	Peru

2.7.12. MaskActionType

The action parameter for select command.

Flag	Value	Description
Assert_Deassert	0	**Matcing :** assert SL or inventoried → A **Not Matcing :** deassert SL or inventoried → B
Assert_DoNothing	1	**Matcing :** assert SL or inventoried → A **Not Matcing :** do nothing
DoNothing_Deassert	2	**Matcing :** do nothing **Not Matcing :** deassert SL or inventoried → B
Negate_DoNothing	3	**Matcing :** negate SL or (A → B, B → A) **Not Matcing :** do nothing
Deassert_Assert	4	**Matcing :** deassert SL or inventoried → B **Not Matcing :** assert SL or inventoried → A
Deassert_DoNothing	5	**Matcing :** deassert SL or inventoried → B **Not Matcing :** do nothing
DoNothing_Assert	6	**Matcing :** do nothing **Not Matcing :** assert SL or inventoried → A
DoNothing_Negate	7	**Matcing :** do nothing **Not Matcing :** negate SL or (A → B, B → A)

2.7.13. MaskTargetType

The target parameter for select command.

Flag	Value	Description
S0	0	Session 0
S1	1	Session 1

S2	2	Session 2
S3	3	Session 3
SL	4	Session Flag

2.7.14. SingulationAlgorithm

Inventory algorithm to use

Flag	Value	Description
FIXEDQ	0	Use start q as a fixed value
DYNAMICQ	1	Use start q, min q, max q dynamically.

2.7.15. **ResultCode**

The results of the action of a method or event.

Flag	Value	Description
NoError	0x0000	No error
OtherError	0x0001	Other error
Undefined	0x0002	Undefined
MemoryOverrun	0x0003	Memory overrun
MemoryLocked	0x0004	Memory locked
InsufficientPower	0x000B	Insufficient power
NonSpecificError	0x000F	Non-Specific error
InvalidResponse	0xE001	Invalid response
InOperation	0xE002	In operation
OutOfRange	0xE003	Out of range
NotConnected	0xE004	Disconnected
InvalidParameter	0xE010	Invalidate parameter
InvalidInstance	0xE100	Invalid instance
FailSendControlPacket	0xEE00	Failed to send control packet
FailReceivePacket	0xEE01	Failed to receive packet
InvalidControlResponse	0xEE02	Invalidate control response packet
UnknownControlResponse	0xEE0F	Unknown control response
InvalidRegisterParameter	0xEE10	Invalidate register parameter
InvalidRegisterResponse	0xEE11	Invalidate register response
UnknownRegisterResponse	0xEE12	Unknown register response
FailSendRegisterPacket	0xEE11	Failed to send register packet
NotSupported	0xEF00	Not Supported
Timeout	0xEFFF	Timeout
HandleMismatch	0xF001	Handle mismatch
CRCError	0xF002	CRC error on tag response
NoTagReply	0xF003	No tag reply
InvalidPassword	0xF004	Invalid password
ZeroKillPassword	0xF005	Zero kill password
TagLost	0xF006	Tag lost
CommandFormatError	0xF007	Command format error
ReadCountInvalid	0xF008	Read count invalid
OutOfRetries	0xF009	Out of retries

ParamError	0xFFFB	Parameter error
Busy	0xFFFC	Busy
InvalidCommand	0xFFFD	Invalid command
LowBattery	0xFFFE	Low battery
OperationFailed	0xFFFF	Operation failed