



ATID Co.,Ltd

ATID SDK Reference Guide for iPhone Developers

ATID SDK Reference Guide

Ryu Eunju
2018-03-09

History of revision

version	Date of revision	Reason for revision ¹	Details of revision ²	Prepared by
v0.1	2017. 4. 21.	draft		Ryu, Eun-ju
v0.2	2017.11.07	addition	Addition of barcode module related API and organization of symbol detail setting param (p.34~ p.38)	Ryu, Eun-ju
V0.3	2018. 3. 9.	addition	Change in project name (ATx88 -> ATID Reader), Addition of 'tagAccessResultWithFreq' (p.57)	Ryu, Eun-ju

¹Reason for revision : enter whether detail of enactment or revision is addition, modification or deletion

²Details of revision :describe page number which revision is made and detail of change

Content

1.	Intro.....	10
2.	EAREader Class.....	11
2.1.	Initialize Methods.....	11
2.1.1.	initWithDevice.....	11
2.1.2.	Disconnect.....	12
2.2.	Reader Properties.....	13
2.2.1.	peripheral.....	13
2.2.2.	getAction.....	13
2.2.3.	setDelegate	13
2.3.	Action Methods.....	14
2.3.1.	Inventory	14
2.3.2.	readMemory.....	14
2.3.3.	writeMemory	15
2.3.4.	lock	15
2.3.5.	unlock.....	16
2.3.6.	permaLock.....	16
2.3.7.	kill.....	17
2.3.8.	stop.....	17
2.3.9.	loadStoredData	18
2.3.10.	deleteAllStoredData	18
2.3.11.	defaultParameter	19
2.3.12.	saveParameter	19
2.4.	Device Properties.....	20

2.4.1.	firmwareVersion	20
2.4.2.	powerGainScope.....	20
2.4.3.	continuousMode	20
2.4.4.	powerGain	21
2.4.5.	operationTime	21
2.4.6.	inventoryTime	21
2.4.7.	IdleTime	22
2.4.8.	autoOffTime.....	23
2.4.9.	accessPassword.....	23
2.4.10.	inventorySession.....	23
2.4.11.	sessionFlag	24
2.4.12.	selectionMask	24
2.4.13.	usedSelectionMask.....	25
2.4.14.	removeSelectionMask	25
2.4.15.	clearSelectionMask	26
2.4.16.	isUseKeyAction	26
2.4.17.	autosaveMode	26
2.4.18.	storedCount.....	26
2.4.19.	reportMode.....	27
2.4.20.	batteryStatus	27
2.4.21.	rssiMode.....	27
2.4.22.	clearEpcMask.....	28
2.4.23.	saveEpcMask	28
2.4.24.	epcMaskCount.....	28

2.4.25.	addEpcMask	28
2.4.26.	getEpcMask.....	29
2.4.27.	epcMaskMatchingMethod	29
2.4.28.	getChannelMask.....	29
2.4.29.	getChannel.....	29
2.4.30.	setChannel.....	30
2.4.31.	getChannelFreq.....	30
2.4.32.	rfiddeviceget.....	30
2.4.33.	linkProfile	30
2.4.34.	setLinkProfile	31
2.4.35.	defaultProfile	31
2.4.36.	setDefaultProfile	31
2.5.	Barcode Methods	32
2.5.1.	setBarcodeMode.....	32
2.5.2.	startScan.....	32
2.5.3.	stopScan.....	33
2.5.4.	setBarcodeParam.....	33
2.5.5.	getBarcodeParam.....	33
2.5.6.	setPropBarcodeMode.....	34
2.5.7.	PropBarcodeMode.....	34
2.5.8.	charsetget.....	34
2.5.9.	charsetset.....	35
2.5.10.	barcodedeviceset	37
2.5.11.	barcodedeviceget.....	38

3.	EADevice BluetoothLe Class.....	39
3.1.	Initialize Methods.....	39
3.1.1.	initWithPeripheral	39
3.2.	Properties.....	40
3.2.1.	delegate.....	40
3.2.2.	name	40
3.2.3.	address.....	40
3.2.4.	peripheral.....	41
3.3.	Methods.....	41
3.3.1.	disconnect	41
3.3.2.	writeData.....	41
4.	EAMinMaxValue.....	42
4.1.	Properties.....	42
4.1.1.	min.....	42
4.1.2.	max.....	42
5.	LockParam	43
5.1.	Properties.....	43
5.1.1.	killPassword	43
5.1.2.	accessPassword.....	43
5.1.3.	epc	43
5.1.4.	tid	44
5.1.5.	user.....	44
6.	EAResultType.....	45
6.1.	Methods.....	45

6.1.1.	msg.....	45
7.	EAResultData.....	46
7.1.	Properties.....	46
7.1.1.	result	46
7.1.2.	mData.....	46
8.	EASelectMaskParam.....	47
8.1.	Initialize Methods.....	47
8.1.1.	initWithIndex.....	47
8.1.2.	initWithParameterIndex.....	48
8.1.3.	initWithParameterLength.....	49
8.2.	Properties.....	50
8.2.1.	index.....	50
8.2.2.	target	50
8.2.3.	action	50
8.2.4.	bank.....	50
8.2.5.	offset.....	51
8.2.6.	mask.....	51
8.2.7.	length	51
8.2.8.	used.....	51
9.	ChannelItem	52
9.1.	Initialize Methods.....	52
9.1.1.	init	52
9.1.2.	initWithSlot	52
9.2.	Properties.....	53

9.2.1.	mSlot.....	53
9.2.2.	mIsUsed.....	53
10.	Deletagate Interfaces.....	54
10.1.	EAREaderDelegate	54
10.1.1.	readerInitialized.....	54
10.1.2.	updateDeviceState.....	54
10.1.3.	readTagResult	55
10.1.4.	changedActionState.....	55
10.1.5.	changedRemoteKey	56
10.1.6.	barcodeScan.....	56
10.1.7.	tagAccessResult	57
10.1.8.	tagAccessResultWithFreq	57
11.	Enumerators	59
11.1.	ResultType	59
11.2.	BankType.....	60
11.3.	SessionType.....	60
11.4.	FlipModeType	60
11.5.	MaskTargetType.....	61
11.6.	MaskActionType.....	61
11.7.	MaskType	62
11.8.	CommandType	62
11.9.	BarcodeType	62
11.10.	ParamName	64
11.10.1.	Zebra(SE4710/SE965)	64



ATID SDK Reference Guide for iPhone Developers

ATID SDK Reference Guide

Company

ATID Co.,Ltd

Doc.

Drafter

EunJu RYU

Date

2018-03-09

Ver.

v0.3

11.10.2. Honeywell(N3680/N6603).....71

1. Intro

The purpose of this document is to describe how to use SDK Library for developers wishing to develop iPhone application program by using RF Prisma SDK Library.

Xcode7.3.1 was used as development tool. Platform to be developed supports iOS 8.3 or higher.

2. EReader Class

EReader Class provides ATID device and Programmable Interface on iPhone. CoreBluetooth framework provided as iOS SDK is used to connect ATID device to an iPhone.

2.1. Initialize Methods

Like ordinary CocoaTouchClass, EReader class also provides a function for initialization. Initialization function beginning with init. provides initWithDevice function and initWithBTDevice function as initialization.

2.1.1. initWithDevice

Method to create EReader entity communicating by using ATID device BLE on iPhone.

Syntax

```
- (id)initWithDevice:(EADevice *)device delegate:(id<EReaderDelegate>)delegate;
```

Parameters

device : designates the Instance of EADevice BluetoothLe Class controlling access information that includes CPeripheral Object of the CoreBluetooth framework that the access is already completed.

delegate : designates the delegate protocol interface to receive information from ATID device via events such as a change in state of EReader.

Remarks

Make sure to designate the Instance of EADevice BluetoothLe Class that the CPeripheral which the access is already completed is included. For access to BLE, use CoreBluetooth Framework. After access, SDK is used to communicate since initialization of ATID device.

2.1.2. Disconnect


Terminates connection to ATID device connected to iPhone.

Syntax

```
- (void)disconnect;
```

Remarks

Disconnect Method calls disconnect of CBPeripheral. After this Method is called, CBPeripheral Instance included as EADevice BluetoothLe Instance is disabled.

		ATID SDK Reference Guide for iPhone Developers					
ATID SDK Reference Guide					Company	ATID Co.,Ltd	
Doc.		Drafter	EunJu RYU	Date	2018-03-09	Ver.	v0.3

2.2. Reader Properties

2.2.1. peripheral

Returns to Instance of CBPeripheral Class used to connect to ATID device.

Syntax

```
- (CBPeripheral *)peripheral;
```

Remarks

Refer to CoreBluetooth framework for Interface to check state of connection to ATID device.

2.2.2. getAction

Returns current active state of EAREader Instance.

Syntax

```
- (CommandType)getAction;
```

Remarks

getAction Method returns Action Command which EAREader transmitted to ATID device last to let you know what task current EAREader commanded to ATID device and what task is being performed.

2.2.3. setDelegate

Changes the recipient that will receive events from Instance of EAREader.

Syntax

```
- (void)setDelegate:(id<EAREaderDelegate>)delegate;
```

Parameters

delegate : designates the Instance of Class that implements the EAREaderDelegate Protocol which can receive an event.

Remarks

If you wish to change recipient that receives an event from EAREader Instance, you can use the setDelegate Method.

2.3. Action Methods

2.3.1. Inventory

Automatically Perform Inventory function depending on the state of ATID device.

Syntax

```
- (ResultType)inventory;
```

Return

Returns a result of execution of an instruction in a form of ResultType.

Remarks

Return NoError when operating inventory method normally. While Inventory method is working, you cannot use another method and thus you stop motion by using stop method and should call another method.

If ATID device reads tag by inventory command, readTagResultMethod of EAREaderDelegate Protocol will be called.

2.3.2. readMemory

readMemory is a command to read Tag memory directly.

Syntax

```
- (ResultType)readMemory:(BankType)bank offset:(int)offset length:(int)length;
```

Parameters

bank : set Memory Bank of data to read.

offset : set initial address of data to read. (in WORD)

length : set the length of data to read.(in WORD)

Return

Return a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Return NoError if operating readMemory method normally. While readMemory method is working, you cannot use another method and thus you stop motion by using stop method and should call another method.

2.3.3. writeMemory

writeMemory is a command to write memory of tag directly.

Syntax

```
- (ResultType)writeMemory:(BankType)bank offset:(int)offset value:(NSString *)value;
```

Parameters

bank : sets Memory Bank of tag to perform data writing.

offset : sets initial address for data to begin writing.(in WORD)

data : enters data to write in Hex value.

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Returns NoError if operating writeMemory method normally. While writeMemory method is working, you cannot use another method and thus you stop motion by using stop method and should call another method.

2.3.4. lock

lock method is used in locking to prevent access to Tag Memory.

Syntax

```
- (ResultType)lock:(LockParam *)param;
```

Parameters

param: lock option of Memory Bank

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

A result of execution of Lock is returned as readTagResult event. If Lock method succeeds, returns as NoError. If Lock method fails, returns as a value other than NoError.

If event handler is called, Lock motion will stop automatically. If you wish to cancel while lock is being performed, you only have to call stop method.

2.3.5. unlock

Unlock method is used in unlocking to prevent access to Tag Memory.

Syntax

```
- (ResultType)unlock:(LockParam *)param;
```

Parameters

param : lock option of Memory Bank

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

A result of execution of unlock is returned as readTagResult event. If unlock method succeeds, returns as NoError. If unlock method fails, returns as a value other than NoError.

If event handler is called, unlock motion will stop automatically. If you wish to cancel while unlock is being performed, you have only to call stop method.

2.3.6. permaLock

Syntax

```
- (ResultType)permaLock:(LockParam *)param;
```

Parameters

param : lock option of Memory Bank

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

ATID SDK Reference Guide					Company		ATID Co.,Ltd	
Doc.		Drafter	EunJu RYU	Date	2018-03-09	Ver.	v0.3	

A result of execution of permaLock is returned as readTagResult event. If permaLock method succeeds, returns as NoError. If permaLock method fails, return as a value other than NoError.

If event handler is called, permaLock motion will stop automatically. If you wish to cancel while permaLock is being performed, you only have to call stop method.

2.3.7. kill

kill method kills to prevent reaction to ATID device.

Syntax

```
- (ResultType)kill:(NSString *)killPassword;
```

Parameters

killpassword : Kills password of tag for kill to apply.

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Tag which Kill method has been applied is impossible to be restored.

2.3.8. stop

stop method stops motions of all Action series.

Syntax

```
- (ResultType)stop;
```

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Cancels all works which ATID device is operating and change state to stop state.

2.3.9. loadStoredData

loadStoredData method returns filter tag list stored in ATID device and barcode scanned while auto save mode operates as readTagResult event and barcodeScan event.

Syntax

```
- (ResultType)loadStoredData;
```

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

readTagResult event or barcodeScan event takes place from ATID device. If all tags or barcodes stored in memory are returned, ResultType will be returned.

2.3.10. deleteAllStoredData

deleteAllStoredData method deletes all data lists stored in the internal memory of the ATID device.

Syntax

```
- (ResultType)deleteAllStoredData;
```

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

If all values stored in internal memory of ATID device are deleted, NoError will be returned.

2.3.11. defaultParameter

defaultParameter method initializes all parameters of ATID device as Default.

Syntax

```
- (ResultType)defaultParameter;
```

Return

Returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Initializes all parameters in ATID device and then return a progress of execution. If it operates normally, NoError will be returned.

2.3.12. saveParameter

This method sets so that reader can remember parameter values even after power is switched off and on or during hardware reboot by saving parameter set in device.

Syntax


```
- (ResultType)saveParameter;
```

Return

Returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

When ATID device operates normally after setting parameter values in ATID device, NoError will be returned.

		ATID SDK Reference Guide for iPhone Developers					
ATID SDK Reference Guide					Company	ATID Co.,Ltd	
Doc.		Drafter	EunJu RYU	Date	2018-03-09	Ver.	v0.3

2.4. Device Properties

2.4.1. firmwareVersion

getFirmwareVersion method returns Firmware version of ATID device as character string.

Syntax

```
- (NSString *)firmwareVersion;
```

Remarks

If Firmware version is returned normally from the ATID device, character string will be returned.

2.4.2. powerGainScope

getPowerGainScope method returns the scope of antenna output level of ATID device.

Syntax

```
- (EAMinMaxValue)powerGainScope;
```

Remarks

If you executed getPowerGainScope method successfully, you can check minimum and maximum value through min, max of EAMinMaxValue. Value returned as min, max is * 10 of output level value. For example, if maximum value is 300 and minimum value is 110, antenna level ranges from 30dBm to 11dBm.

2.4.3. continuousMode

continuousMode property returns or sets whether to read tag once or consecutively when ATID device performs Inventory function.

Syntax

```
@property (nonatomic, assign) BOOL continuousMode;
```

Remarks

This function returns or sets whether to read tag once or consecutively during Inventory operation. If this value is YES, it means Continuous Mode has been set. If this value is NO, it means Continuous Mode has not been set.

2.4.4. powerGain

powerGain method returns or sets Power Gain of ATID device.

Syntax

```
@property (nonatomic, assign) int powerGain;
```

Remarks

powerGain property is integer type. Value *10 of power output value is set or returned.

For example, if antenna power is 30dBm, value of 300 will be set or returned.

2.4.5. operationTime

operationTime property sets or returns operating time of ATID device.

Syntax

```
@property (nonatomic, assign) int operationTime;
```

Remarks

If operationTime value is set to 0, it will continue to operate until stop method is called.

Unit of setting operationTime value is ms.

2.4.6. inventoryTime

inventoryTime property sets or returns time that antenna actually operates when ATID device performs Inventory.

Syntax

```
@property (nonatomic, assign) int inventoryTime;
```

Remarks

While Inventory is operated, Reader performs Inventory during Inventory Time. Repeats operation in such a manner that it idles during Read Idle Time and it operates during Inventory Time.

ATID SDK Reference Guide					Company		ATID Co.,Ltd	
Doc.		Drafter	EunJu RYU	Date	2018-03-09	Ver.	v0.3	

2.4.7. IdleTime

idleTime property returns or sets idle time when ATID device performs Inventory.

Syntax

```
@property (nonatomic, assign) int idleTime;
```

Remarks

This function is used to minimize interference among readers in an environment where several ATID devices are operated and prevent overheating of reader resulting from long full time operation. In the event of RF Channel Hopping, during Power Idle Time(ms), reader powers off and during Inventory Time(ms), reader powers on.

*Note

Inventory Time and Idle Time vary in a range of value to be applied depending on Global.

-Inventory Time

Japan: 50 ~ 4000ms

Europ: 50 ~ 4000ms

Others: 50 ~ 400ms(sum with Idle time cannot exceed 400ms)

-Idle Time

Japan: 50 ~ 4000ms

Europ: 100 ~ 4000ms

Others: 0 ~ 400ms(sum with Inventory time cannot exceed 400ms)

2.4.8. autoOffTime

autoOffTime property sets or returns time that ATID device turns off automatically when it does not operate.

Syntax

```
@property (nonatomic, assign) int autoOffTime;
```

Remarks

This function allows reader to turn off automatically if ATID device is not used for a certain time to save Battery to device. Unit of setting autoOffTime is sec. If you do not want to use Auto Off Time function, you have only to set this value to 0.

2.4.9. accessPassword

This function sets or returns Access Password needed when you perform Access Command for tag that lock is placed on ATID device.

Syntax

```
@property (nonatomic, strong) NSString *accessPassword;
```

Remarks

When implementing Read/Write/Lock Command that Access Password is set to 0, Access Password is not used. Password of 8Byte unit is set. Access Password is used.

2.4.10. inventorySession

This function sets or returns session of tag to access when ATID device performs Inventory.

Syntax

```
@property (nonatomic, assign) SessionType inventorySession;
```

Remarks

This function performs Inventory with information on tag session state related to inventory motion while ATID device performs Inventory. inventorySession property designates session that ATID device accesses tag when it performs Inventory.

2.4.11. sessionFlag

This function sets or returns session state value subject to inventory among session states of tag that accessed while ATID device performs Inventory.

Syntax

```
@property (nonatomic, assign) SessionFlag sessionFlag;
```

Remarks

If sessionFlag is A or B, Inventory is performed regardless of Session state. If sessionFlag is set to A Only or B Only, only tag with designated session state value is inventoried.

2.4.12. selectionMask

This function sets or returns Selection Mask so that ATID device can inventory only tag with specific condition when it performs Inventory. ATID device has eight Selection Mask slots (0~7). In other words, up to eight tags is possible for Selection Mask.

Syntax

```
@property (nonatomic, strong) EASelectMaskParam*selectionMask;
```

Remarks

This function returns or sets arrangement pointer of instance of ATSelectionMaskParam Class wrapping Selection Mask information which meets Gen2 rule.

2.4.13. useSelectionMask

useSelectionMask property sets or returns whether to perform Inventory by using Selection Mask information which was set during Inventory motion.

Syntax

```
@property (nonatomic, assign) BOOL useSelectionMask;
```

Remarks

If this value is set to NO, Inventory method performs Inventory on all general tags. If this value is set to YES, Inventory method inventories only tag which has been filtered to Selection Mask information that has been set to selection Mask property.

2.4.14. removeSelectionMask

removeSelectionMask method deletes Selection Mask information on specific index stored in ATID device.

Syntax

```
- (void)removeSelectionMask:(int)index;
```

Parameters

index :Mask index parameter

Remarks

Deletes Selection Mask information on designated Index.

2.4.15. clearSelectionMask

clearSelectionMask method deletes all Selection Mask information in ATID device.

Syntax

```
- (void)clearSelectionMask;
```

2.4.16. isUseKeyAction

This function returns or sets whether to use Hardware Key in ATID device.

Syntax

```
@property (nonatomic, assign) BOOL isUseKeyAction;
```

Remarks

If this property is set to YES, Hardware Key in ATID is used. If this property is set to NO, H/W Key cannot be used.

2.4.17. autosaveMode

This function returns or set whether to store data detected in embedded memory while ATID device performs Inventory or Barcode Scan.(This function is not supported in ATS100)

Syntax

```
@property (nonatomic, assign) BOOL autosaveMode;
```

Remarks

If autosaveMode is set to YES, ATID device stores tag information inventoried or barcode information scanned in embedded memory.

2.4.18. storedCount

This function returns the number of data stored in embedded memory of ATID device. (This function is not supported in ATS100)

Syntax

```
- (int)storedCount;
```

2.4.19. reportMode

This function returns or set whether to generate read tag event when ATID device reads tag data stored in memory while performing Inventory.

Syntax

```
@property (nonatomic, assign) BOOL reportMode;
```

Remarks

If report mode is YES and tag value stored in memory is read, read tag event does not occur. If report mode is NO and tag value stored in memory is read, read tag event occurs.

2.4.20. batteryStatus

This function returns residual quantity of battery to ATID device. Return value is from 0 to 4. If return value is 4, this means battery is being charged or in a status of maximum charge.

Syntax

```
- (int)batteryStatus;
```

2.4.21. rssiMode

rssiMode returns or sets whether to report RSSI value when ATID device reads tag value.

Syntax

```
@property (nonatomic, assign) BOOL rssiMode;
```

Remarks

If RSSI Mode is set to YES, tag value and RSSI value are returned at the same time as readTagResult event. If RSSI Mode is set to NO, only Tag value is returned. Default value is NO.

2.4.22. clearEpcMask

This function deletes all memory EPC masks used when ATID device performs Inventory.

Syntax

```
- (ResultType)clearEpcMask;
```

2.4.23. saveEpcMask

This function applies added memory EPC mask to ATID device.

Syntax

```
- (ResultType)saveEpcMask;
```

2.4.24. epcMaskCount

This function returns the number of memory EPC mask stored in ATID device.

Syntax

```
- (int)epcMaskCount;
```

2.4.25. addEpcMask

This function adds memory EPC mask when ATID device performs Inventory.

Syntax

```
- (ResultType)addEpcMask:(int) offset length:(int)length mask:(NSString *)mask;  
- (ResultType)addEpcMask:(EASelectMaskePCParam *)masks;
```

Parameters

offset : designates initial address which a mask begins as bit. This should be equal to 16 bit or greater than 16bit.

length : sets the length which a mask will be applied in bit.

mask : designates value to be masked as HEX type string.

masks : EASelectMaskePCParam class entity arrangement pointer that contains mask information

ATID SDK Reference Guide					Company		ATID Co.,Ltd	
Doc.		Drafter	EunJu RYU	Date	2018-03-09	Ver.	v0.3	

2.4.26. getEpcMask

This function returns EPC mask information located in index given among memory EPC mask values stored in ATID device.

Syntax

```
- (EASelectMaskEPCParam *)getEpcMask:(int)index;
```

Parameters

index : cardinal number in which memory EPC mask is stored.

2.4.27. epcMaskMatchingMethod

This function returns matching status of whether to filter a tag that EPC mask stored in ATID device accords or to filter a tag that EPC mask stored in ATID device does not accord.

Syntax

```
@property (nonatomic, assign) BOOLepcMaskMatchMethod;
```

Remarks

If epcMaskMatchingMethod is set to YES, a value which accords with set EPC mask is masked. If epcMaskMatchingMethod is set to NO, a value which does not accord with set EPC mask is masked.

2.4.28. getChannelMask

getChannelMask method is to return arrangement of mask value of frequency table information.

Syntax

```
- (NSArray *)getChannelMask;
```

2.4.29. getChannel

getChannel method is to return frequency table information set on ATID device.

Syntax

```
- (NSArray *)getChannel;
```

2.4.30. setChannel

setChannel method is to set frequency table information on ATID device.

Syntax

```
- (void)setChannel:(NSArray *)table;
```

Parameters

Table: frequency Table setting parameter

2.4.31. getChannelFreq

getChannelFreq method is to return frequency information on ATID device.

Syntax

```
- (NSString *)getChannelFrequency:(int)slot;
```

2.4.32. rfiddeviceget

rfiddeviceget method is to return whether UHF RFID module has been applied to ATID device connected at present.

Syntax

```
- (int)rfiddeviceget;
```

2.4.33. linkProfile

linkProfile method is to return link profile information on ATID device.

Syntax

```
- (int)linkProfile;
```

2.4.34. setLinkProfile

setLinkProfile method is to set link profile on ATID device.

Syntax

```
- (void) setLinkProfile:(int)linkprofile;
```

Parameters

linkprofile: link profile setting parameter

2.4.35. defaultProfile

defaultProfile method is to return default link profile information on ATID device.

Syntax

```
- (int)defaultProfile;
```

2.4.36. setDefaultProfile

setDefaultProfile method is to set default link Profile of ATID device.

Syntax

```
- (void) setDefaultProfile:(int)defaultprofile;
```

Parameters

defaultprofile: default link profile setting parameter

2.5. Barcode Methods

2.5.1. setBarcodeMode

setBarcodeMode method is to set so that ATID device can use Barcode Scanner.

Syntax

```
- (ResultType)setBarcodeMode:(BOOL)enabled;
```

Parameter

enabled : BOOL type determining whether to enable Barcode Mode

Remarks

If setBarcodeMode is called YES, ATID device cannot implement UHF related command and only barcode related command is implemented. If you wish to implement UHF related command, you have only to call enabled NO.

2.5.2. startScan

This method is for ATID device to scan barcode in Barcode Mode.

Syntax

```
- (ResultType)startScan;
```

Remarks

If startScan method operates normally, Aimer is launched from scanner, option device.

To stop Barcode Scan, you have only to use stopScan method.

If Barcode is scanned, barcode information scanned through barcodeScan of EAREaderEventListener is returned. Even if Barcode Scan fails, No Read is returned through barcodeScan method.

2.5.3. stopScan

stopScan method is to stop Barcode Scan in Barcode Mode.

Syntax

```
- (ResultType)stopScan;
```

Remarks

Stops barcode scan compulsorily while barcode module, option device of ATID scans barcode.

2.5.4. setBarcodeParam

setBarcodeParam method is to store set value in Barcode Module.

Syntax

```
- (ResultType)setBarcodeParam:(NSArray *)paramData;
```

Parameters

paramData : pointer of NSArray entity which stores Barcode set value

Remarks

setBarcodeParam method receives set value of Barcode Module in a form of bite arrangement. Transmits wrapping setting bite arrangement of BarcodeOption that you want to set in NSArray entity.

2.5.5. getBarcodeParam

getBarcodeParam method is to return set value which has been set in Barcode Module.

Syntax

```
- (NSArray *)getBarcodeParam:(NSArray *)paramData;
```

Parameters

paramData : pointer of NSArray entity which names of set value that you want to return are stored

Remarks

getBarcodeParam method is to check set value stored in Barcode Module. This method returns bite arrangement in which names and values are stored in order as NSArray

entity by transmitting after wrapping code bite arrangement representing barcode parameter name as NSArray entity with paramData.

2.5.6. setPropBarcodeMode

setPropBarcodeModesms method is to set operating mode of a device.

Syntax

```
- (void)setPropBarcodeMode:(int)mode;
```

Parameters

mode : operating mode value of a device (0: Inventory 1: Barcode)

Remarks

setPropBarcodeMode method receives operating mode of a device in a form of int.
Transmits by wrapping eigen value of motion that you want to set in int entity.

2.5.7. PropBarcodeMode

PropBarcodeMode method is to return operating mode of a device at present.

Syntax

```
- (int)PropBarcodeMode;
```

Remarks

PropBarcodeMode method is to check operating mode of a device connected at present and returns representative value of mode according to motion as int.

2.5.8. charsetget

charsetget method is to return set value of character set which decodes barcode value at present.

Syntax

```
- (int)charsetget;
```

Remarks

Charsetget method is to check language setting which decodes barcode value and returns index value of language stored as NSArray.

2.5.9. charsetset

charsetset method is to set character set which decodes barcode value.

Syntax

```
- (void)charsetset: (int)text;
```

Parameters

text: index of NSArray which set values of decoding language which you want to set are stored.

Remarks

Charsetset method is to set a language which decodes barcode value and receives index value of language wrapped as NSArray entity. char set which you can set is as follows and index value as shown below.

*Char Setlist

Char
Western (ASCII)
Western (Mac OS Roman)
Japanese (Mac OS)
Traditional Chinese (Mac OS)
Korean (Mac OS)
Greek (Mac OS)
Cyrillic (Mac OS)
Thai (Mac OS)
Simplified Chinese (Mac OS)
Central European (Mac OS)
Symbol (Mac OS)
Dingbats (Mac OS)
Turkish (Mac OS)
Croatian (Mac OS)
Icelandic (Mac OS)
Romanian (Mac OS)
Unicode™ (UTF-16)

Unicode™ (UTF-8)

Western (ISO Latin 1)

Central European (ISO Latin 2)

Western (ISO Latin 3)

Central European (ISO Latin 4)

Cyrillic (ISO 8859-5)

Arabic (ISO 8859-6)

Greek (ISO 8859-7)

Hebrew (ISO 8859-8)

Turkish (ISO Latin 5)

Nordic (ISO Latin 6)

Thai (ISO 8859-11)

Baltic Rim (ISO Latin 7)

Celtic (ISO Latin 8)

Western (ISO Latin 9)

Latin-US (DOS)

Greek (DOS)

Baltic Rim (DOS)

Western (DOS Latin 1)

Central European (DOS Latin 2)

Turkish (DOS)

Icelandic (DOS)

Arabic (DOS)

Cyrillic (DOS)

Thai (Windows, DOS)

Japanese (Windows, DOS)

Simplified Chinese (Windows, DOS)

Korean (Windows, DOS)

Traditional Chinese (Windows, DOS)

Western (Windows Latin 1)

Central European (Windows Latin 2)

Cyrillic (Windows)

Greek (Windows)

Turkish (Windows Latin 5)

Hebrew (Windows)

Arabic (Windows)

Baltic Rim (Windows)

Vietnamese (Windows)

Chinese (GB 18030)

Japanese (ISO 2022-JP)

Korean (ISO 2022-KR)

Japanese (EUC)

Simplified Chinese (EUC)

Korean (EUC)

Japanese (Shift JIS)

Cyrillic (KOI8-R)

Traditional Chinese (Big 5)

Western (Mac Mail)

Traditional Chinese (Big 5 HKSCS)

Western (NextStep)

Non-lossy ASCII

2.5.10. barcodedevicesset

barcodedevicesset method is to store Barcode Module.

Syntax

```
- (void)barcodedevicesset: (int)model;
```

Parameters

model : value numbered according to types of Barcode module

Remarks

Barcodedeviceset method receives Barcode Module in a form of int. Barcodedeviceset method transmits eigen value of Barcodemodule that you want to set by wrapping it in int entity.(cf. module 0: SE4710 1:N3680/N6603 2:SE965)

2.5.11. barcodedeviceget

barcodedeviceget method is to return Barcode Module installed on a device connected at present.

Syntax

```
- (int)getbarcodedevice;
```

Remarks

Barcodedeviceget method is to check Barcode Module installed on a device connected at present and returns representative value representing model of module as int value.

3. EADevice BluetoothLe Class

EADevice BluetoothLe Class plays a role to wrap and deliver CBPeripheral Class Instance to EAREader Class after connecting ATID tool to BLE tool.

3.1. Initialize Methods

3.1.1. initWithPeripheral

Syntax

```
- (id)initWithPeripheral:(CBPeripheral *)peripheral delegate:(id<EADeviceInitializeDelegate>)callback;
```

Parameters

Peripheral: it designates EADevice BluetoothLe Class Instance which manages the linking information including CPeripheral Object of CoreBluetooth framework already completely linked.

Delegate: it designates the delegate protocol interface aiming at receiving information from ATID tool through events such as the status change of EAREader, and etc.

Remarks

Make sure to specify the EADevice BluetoothLe Class Instance including the already linked CPeripheral Object. As for BLE linkage, use CoreBluetooth Framework and communicate by using this SDK from the initialization of ATID tool after being linked.

ATID SDK Reference Guide					Company		ATID Co.,Ltd	
Doc.		Drafter	EunJu RYU	Date	2018-03-09	Ver.	v0.3	

3.2. Properties

3.2.1. delegate

When receiving data through BLE communication, set an agent to deal with the received data or retrocede it.

Syntax

```
@property (weak, nonatomic) id<EADevice ReadDataDelegate> delegate;
```

Remarks

Set the object actualized EADevice ReadDataDelegate interface, which is performing the receiving routine, to take care of the receipt of data.

3.2.2. name

Retrocede the name of BLE Device of ATID tool connected to the Host tool.

Syntax

```
- (NSString *)name;
```

Remarks

Retrocede the name of BLE Device of ATID tool connected to the Host tool.

3.2.3. address

Retrocede the BLE Device Address of ATID tool connected to the Host tool.

Syntax

```
- (NSString *)address;
```

Remarks

Retrocede the temporary MAC Address of BLE Device of ATID tool connected to the Host tool.

3.2.4. peripheral

Retrocede CBPeripheral Clas Instance used for connecting ATID tool.

Syntax

```
- (CBPeripheral *)peripheral;
```

Remarks

You may refer to the CBPeripheral Object of Cocoa Touch for providing the detail information of BLE linkage connected to ATID tool.

3.3. Methods

3.3.1. disconnect

It terminates the BLE module linkage connected to ATID tool.

Syntax

```
- (void)disconnect;
```

Remarks

Terminate the linkage with ATID tool and nullify CBPeripheral Object.

3.3.2. writeData

It transmits data through BLE to ATID tool.

Syntax

```
- (void)writeData:(NSData *)data;
```

Parameter

data: NSData object having data intending to transmit

Remarks

Transmit data by using BLE of ATID tool through writeData method and receive the received data through EADevice ReadDataDelegate Interface.

4. EAMinMaxValue

4.1. Properties

4.1.1. min

Syntax

```
int min;
```

Remarks

Storing minimum value

4.1.2. max

Syntax

```
int max;
```

Remarks

Storing maximum value

5. LockParam

5.1. Properties

5.1.1. killPassword

Set whether the scope of Kill Password is controlled or retrocede it

Syntax

```
@property (nonatomic) BOOL killPassword;
```

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of Kill Password as the subject.

5.1.2. accessPassword

Set whether Access Password scope is controlled or retrocede it.

Syntax

```
@property (nonatomic) BOOL accessPassword;
```

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of Access Password as the target.

5.1.3. epc

Set whether EPC memory is controlled or retrocede it.

Syntax

```
@property (nonatomic) BOOL epc;
```

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of EDC memory as the target.

5.1.4. tid

Set whether TID memory is controlled or retrocede it.

Syntax

```
@property (nonatomic) BOOL tid;
```

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of TID memory as the target.

5.1.5. user

Set whether User memory is controlled or retrocede it.

Syntax

```
@property (nonatomic) BOOL user;
```

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of User memory as the target.

6. EAResultType

Utility Class is for converting message strings to the enumeration of ResultType.

6.1. Methods

6.1.1. msg

Syntax

```
+(NSString *)msg:(ResultType)code;
```

Parameters

Code: ResultType enumeration to be converted a message string

Remarks

Convert message strings of ResultType

7. EAResultData

ResultType Return Class is used if method is to retrocede ResultType and data.

7.1. Properties

7.1.1. result

Syntax

```
ResultType mResult;
```

Remarks

Show the retroceded ResultType.

7.1.2. mData

Syntax

```
NSData * mData;
```

Remarks

NSData object showing the retroceded data.

8. EASelectMaskParam

EASelectMaskParam Class an Interface Class which sets Selection Mask whose size is Gen2 in ATID tool or retrocedes it. In ATID tool, you can set up to 8 Selection Masks.

8.1. Initialize Methods

8.1.1. initWithIndex

It generates the basic EASelectMaskParam object by the radix of Selection Mask.

Syntax

```
- (id)initWithIndex:(int)index;
```

Parameters

Index: it is an integer indicating the ordinance of Selection Mask whose value is from 0 to 7.

Remarks

initWithIndex methods generate EASelectMaskParamClass Instance initialized by the basic Selection Mask value with index value.

8.1.2. initWithParameterIndex

It generates the EASelectMaskParam object comprises the information of Selection Mask with the given value of Parameter

Syntax

```
- (id)initWithParameterIndex:(int)index target:(MaskTargetType)maskTarget
action:(MaskActionType)maskAction bank:(BankType)maskBank offset:(int)maskOffset
mask:(NSString *)maskData used:(BOOL)usedMask;
```

Parameters

Index: it is an integer indicating the ordinance of Selection Mask whose value is from 0 to 7.

Mask Target: MaskTargetType enumeration indicating Tag session which Selection Mask is applied to.

Mask Action: MaskTargetType enumeration indicating the state of Tag session after Selection Mask is applied to.

Mask Bank: BankType enumeration indicating Tag memory bank to which Selection Mask is compared.

maskOffset: it is integer by bit indicating the start address in which Selection begins comparing Selection Mask data.

maskData: it is NSString in the type of Hex indicating Mask data which is the target of being compared to Selection Mask.

usedMask: it is BOOL type indicating whether the generated EASelectMaskParam object will be used as Selection Mask in ATID tool..

Remarks

Generate an EASelectMaskParam object with the given information of Parameter. The thing you shall notice is offset is designated by bit. If you want to designate part of PC as the start address of Selection Mask, designate the start address as 16 since 1Word at the place of it in EPC memory bank is the value of PC. As for the data becoming as Selection Mask, compare the given entire Mask data by length.

8.1.3. initWithParameterLength

It generates an EASelectMaskParam object comprising Selection Mask information with given Parameter.

Syntax

```
- (id)initWithParameterLength:(int)index target:(MaskTargetType)maskTarget  
action:(MaskActionType)maskAction bank:(BankType)maskBank offset:(int)maskOffset  
mask:(NSString *)maskData length:(int)maskLength used:(BOOL)usedMask;
```

Parameters

Index: it is an integer indicating the ordinance of Selection Mask whose value is from 0 to 7.

maskTarget: MaskTargetType enumeration indicating Tag session which Selection Mask is applied to.

maskAction: MaskTargetType enumeration indicating the state of Tag session after Selection Mask is applied to.

maskBank: BankType enumeration indicating Tag memory bank to which Selection Mask is compared.

maskOffset: it is an integer by bit indicating the start address in which Selection begins comparing Selection Mask data.

maskData: it is NSString in the type of Hex indicating Mask data which is the target of being compared to Selection Mask.

maskLength: it is an integer by bit indicating the length to compare Mask data which is the target of comparison.

usedMask: it is BOOL type indicating whether the generated EASelectMaskParam object will be used as Selection Mask in ATID tool.

Remarks

Generate a EASelectMaskParam object with the given information of Parameter. The thing you shall notice is the length of Mask is designated by bit. If you want to designate part of PC as the start address of Selection Mask, designate the start address as 16 since 1Word at the place of it in EPC memory bank is the value of PC. This method can be used to designate part of the given Mask data.

8.2. Properties

8.2.1. index

It retrocedes the radix indicating the ordinance of Selection Mask.

Syntax

```
- (int)index;
```

8.2.2. target

It sets or retrocedes whether Tag session to which Selection Mask is applied.

Syntax

```
@property (nonatomic) MaskTargetType target;
```

8.2.3. action

It sets or retrocedes the state of Tag session after Selection Mask is applied.

Syntax

```
@property (nonatomic) MaskActionType action;
```

8.2.4. bank

It sets or retrocedes Tag memory bank in which Selection Mask data is compared.

Syntax

```
@property (nonatomic) BankType bank;
```

8.2.5. offset

It sets the start address by bit which begins comparing Selection Mask data or retrocede.

Syntax

```
@property (nonatomic) int offset;
```

8.2.6. mask

It sets whether Selection Mask data, which is the target of comparison, as Hex type strings or retrocede it.

Syntax

```
@property (strong, nonatomic) NSString *mask;
```

8.2.7. length

It sets the length in which Mask data, which is the target of comparison, is compared or retrocedes it.

Syntax

```
@property (nonatomic) int length;
```

8.2.8. used

It sets whether the Selection Mask in ATID tool is employed or retrocedes it.

Syntax

```
@property (nonatomic) BOOL used;
```

9. ChannelItem

9.1. Initialize Methods

9.1.1. init

It generates a ChannelItem object.

Syntax

```
-(id)init;
```

9.1.2. initWithSlot

It generates an initialized ChannelItem object with the given Parameter.

Syntax

```
-(id)initWithSlot:(int)slot isUsed:(BOOL)isUsed;
```

Parameters

Slot: it is an integerType indicating frequency slot which designates the frequency of channel frequency table.

isUsed: it is a BoolType indicating whether the frequency of the designated slot at the channel frequency table is used.

Remarks

Generate a ChannelItem object by designating slot number and whether it is used or not.

9.2. Properties

9.2.1. mSlot

It sets the radix indicating frequency location of the channel frequency table or retrocede it.

Syntax

```
@property (nonatomic) int mSlot;
```

9.2.2. mIsUsed

It sets whether the frequency of the designated slot at the channel frequency table is used or not or retrocede it.

Syntax

```
@property (nonatomic) BOOL mIsUsed;
```

10. Deletagate Interfaces

10.1. EAREaderDelegate

10.1.1. readerInitialized

It is an interface method to notice the completion of initializing an EAREader object.

Syntax

```
- (void)readerInitialized:(EAREader *)reader;
```

Parameters

reader: a changed EAREader object completely initialized

Remarks

It is called when the initialization of EAREader object connected to ATID tool is complete.

10.1.2. updateDeviceState

It is an interface method to notice the termination of asynchronous work of operational method such as loadStoredTag, saveStoredTag, removeAllStoredTags, or etc.

Syntax

```
- (void)updateDeviceState:(ResultType)error;
```

Parameters

Error: it is a ResultType indicating the consequence of the updated state of ATID tool.

Remarks

It is called when the work accessing to Stored Tag memory is ended to notice the consequence of the work asynchronously as it takes a long work time.

10.1.3. readTagResult

It is an interface method to retrocede the EPC data of the tag read while conducting inventory over ATID tool.

Syntax

```
- (void)readTagResult: (NSString *)tag rssi:(float)rssi phase:(float)phase;
```

Parameters

Tag: string in Hex type of PC and EPC data of the tag read by ATID tool.

rssi: rssi when ATID tool reads the corresponding tag

phase: phase when ATID tool reads tag

Remarks

Call inventory method and an event will be called when ATID tool reads tag.

10.1.4. changedActionState

It is an interface method to notice when the operation of ATID tool is changed..

Syntax

```
- (void)changedActionState:(CommandType)action;
```

Parameters

Action: it is a CommandType enumeration indicating the current operational state of ATID tool.

Remarks

The operational state of ATID tool is changed when method such as inventory, readMemory, writeMemory, lock, unlock, permaLock, stop, and etc. An event is called to notify this to the user.

10.1.5. changedRemoteKey

It is an interface method to notice a change by user's manipulation on Action Key in ATID tool.

Syntax

```
- (void)changedRemoteKey:(int)state;
```

Parameters

State: it is an integerType indicating the state of Remote Key.

Remarks

An event is called with 1 as the value of the state when pressing on Action Key in ATID tool and another event is called with 0 as the value of the state when pressing off Action Key.

10.1.6. barcodeScan

It is an interface method to notice when ATID tool scans barcode in barcode mode.

Syntax

```
- (void)barcodeScan:(BarcodeType)barcodeType codeId:(NSString *)codeId barcode:(NSString *)barcode;
```

Parameters

barcodeType: an enumerationType indicating the type of scanned barcode.

codeId: an string indicating scanned barcode code ID

barcode: ASCII string indicating scanned barcode data

Remarks

An event which calls startScan method when ATID tool entered into Barcode Mode, and which is called when Barcode Scanner, which is option device, reads barcode; and barcode data retrocedes as ASCII string after automatically decoded in inside ATID tool.

10.1.7. tagAccessResult

It is an interface method to notice the consequence of tag accessed by ATID tool.

Syntax

```
- (void)tagAccessResult:(ResultType)error actionState:(CommandType)action epc:(NSString *)epc data:(NSString *)data;
```

Parameters

error: it is a ResultType indicating the result of the method relating to access operation is completed.

action: it is a CommandType enumeration indicating the consequence of a type of operation of ATID tool.

epc: it is a HexType string indicating the EPC value showing the tag accessed by ATID tool.

data: it is a HexType string to retrocede the data read in the Tag in case of Tag Access in which reads data in tag such as readMemory method.

rsi: rsi when ATID tool accesses to the corresponding Tag memory

phase: phase when ATID tool accesses to Tag memory

Remarks

tagAccessResult event is called as the result of accessing to tag in regard to tag access method such as readMemory, writeMemory, lock, unlock, permaLock, and etc., except inventory method. It also includes the data read in tag as the result of readMemory method.

10.1.8. tagAccessResultWithFreq

it is an interface method to notice the result of tag accessed by ATID tool such as tagAccessResult; also, it is a method to convey frequency value in tagAccessResult method.

Syntax

```
- (void)tagAccessResultWithFreq:(ResultType)error actionState:(CommandType)action epc:(NSString *)epc data:(NSString *)data rsi:(float)rsi phase:(float)phase freq:(float)freq;
```

Parameters

ATID SDK Reference Guide					Company		ATID Co.,Ltd	
Doc.		Drafter	EunJu RYU	Date	2018-03-09	Ver.	v0.3	

error: it is a ResultType indicating the result of the method relating to Access operation is completed.

actionState: it is a CommandType enumeration indicating the consequence of the type of operation of ATID tool.

epc: it is a HexType string indicating the EPC value showing the tag accessed by ATID tool.

data: it is a HexType string to retrocede the data read in the Tag in case of Tag Access in which reads data in tag such as readMemory method.

rss: rssi when ATID tool accesses to the corresponding Tag memory

phase: phase when ATID tool accesses to Tag memory

freq: frequency when ATID tool accesses to Tag memory

Remarks

The tagAccessResult event is called as result of the Tag access for Tag access methods such as readMemory, writeMemory, lock, unlock, and permaLock except the inventory method.

Note that data read from the Tags as results of the readMemory method are contained here as well. As for the corresponding method, support status differs depending on Firmware versions of AT188 and AT388 but in case of ATS100, support for all versions is available (AT188: 5.2.2.0 and above, AT388: 2.2.1.1 and above)

11. Enumerators

11.1. ResultType

It shows the operational results of EAREader command and properties.

Flag	Value	Description
ResultNoError	0x0000	Success of result
ResultOtherError	0x0001	Error due to unknown reasons
ResultUndefined	0x0002	Error no defined
ResultMemoryOverrun	0x0003	Going off the range of accessible memory
ResultMemoryLocked	0x0004	Memory locked
ResultInsufficientPower	0x000B	Lack of power
ResultNonSpecificError	0x000F	Trivial error.
ResultInOperation	0xE000	In operation
ResultOutOfRange	0xE001	Out of range
ResultNotConnected	0xE100	Not connected to device.
ResultInvalidParameter	0xE200	Improper transmission of parameter
ResultInvalidResponse	0xE300	Retrocession of wrong answer
ResultNotSupportFirmware	0xEE00	Not supported firmware
ResultTimeout	0xEFFF	Excess of accessible time
ResultHandleMismatch	0xF001	Handle mismatch
ResultCRCError	0xF002	CRC error on tag response
ResultNoTagReply	0xF003	No Tag Reply
ResultInvalidPassword	0xF004	Invalid password
ResultZeroKillPassword	0xF005	Zero kill password
ResultTagLost	0xF006	Tag lost
ResultCommandFormatError	0xF007	Command format error
ResultReadCountInvalid	0xF008	Read count invalid
ResultOutOfRetries	0xF009	Out of retries
ResultParamError	0xFFFFB	Parameter error
ResultBusy	0xFFFFC	Busy
ResultInvalidCommand	0xFFFFD	Invalid Command.

ResultLowBattery	0xFFFE	Low Battery
ResultOperationFailed	0xFFFF	Operation failed

11.2. BankType

It shows Tag memory bank accessible in EASReader.

Flag	Value	Description
Bank_Reserved	0	Reserved memory bank
Bank_EPC	1	EPC memory bank
Bank_TID	2	TID memory bank
Bank_User	3	User memory bank

11.3. SessionType

It shows tag session which is the subject when EASReader conducts inventory.

Flag	Value	Description
Session_S0	0	inventoried S0
Session_S1	1	inventoried S1
Session_S2	2	inventoried S2
Session_S3	3	inventoried S3

11.4. FlipModeType

It shows flag state of the subject session of Selection Mask in EASReader.

Flag	Value	Description
FlipMode_AB	0	A or B
FlipMode_A	1	A only
FlipMode_B	2	B only

11.5. MaskTargetType

It shows the subject session of Selection Mask in EAREader.

Flag	Value	Description
MaskTarget_S0	0	inventoried S0
MaskTarget_S1	1	inventoried S1
MaskTarget_S2	2	inventoried S2
MaskTarget_S3	3	inventoried S3
MaskTarget_SL	4	Selection Flags

11.6. MaskActionType

It shows Action state of Selection Mask in EAREader.

Flag	Value	Description
MaskAction_AB	0	Tag Matching : assert SL or inventoried → A Tag Not-Matching : deassert SL or inventoried → B
MaskAction_AN	1	Tag Matching : assert SL or inventoried → A Tag Not-Matching : do nothing
MaskAction_NB	2	Tag Matching : do nothing Tag Not-Matching : deassert SL or inventoried → B
MaskAction_MN	3	Tag Matching : negate SL or (A → B, B → A) Tag Not-Matching : do nothing
MaskAction_BA	4	Tag Matching : deassert SL or inventoried → B Tag Not-Matching : assert SL or inventoried → A
MaskAction_BN	5	Tag Matching : deassert SL or inventoried → B Tag Not-Matching : do nothing
MaskAction_NA	6	Tag Matching : do nothing Tag Not-Matching : assert SL or inventoried → A
MaskAction_NM	7	Tag Matching : do nothing Tag Not-Matching : negate SL or (A → B, B → A)

11.7. MaskType

It shows Selection Mask method in EReader.

Flag	Value	Description
MaskType_Selection	0	Mask Selection method for Gen2 size
MaskType_EPC	1	EPC Mask Selection method

11.8. CommandType

It shows what is done in asynchronous command of EReader.

Flag	Value	Description
CommandInventory	0x66	Under inventory
CommandReadMemory	0x72	Under working on Read Memory
CommandWriteMemory	0x77	Under working on Write Memory
CommandKill	0x6B	Under working on Kill Tag
CommandLock	0x6C	Under working on Lock
CommandUnlock	0x6D	Under working on Unlock
CommandPermaLock	0x70	Under working on PermaLock
CommandStop	0x73	Cease of operation

11.9. BarcodeType

Flag	Description
BarcodeTypeNoRead	No Read
BarcodeTypeAustralianPost	Australian Post
BarcodeTypeAztecCode	Aztec Code
BarcodeTypeBooklandEAN	Aztec Rune Code
BarcodeTypeBritishPost	Bookland EAN
BarcodeTypeCanadianPost	British Post
BarcodeTypeChinaPost	Canadian Post
BarcodeTypeCodabar	China Post
BarcodeTypeCodablockF	Codabar

BarcodeTypeCode11	Codablock F
BarcodeTypeCode128	Code 11
BarcodeTypeCode16K	Code 128
BarcodeTypeCode32	Code 16K
BarcodeTypeCode39	Code 32
BarcodeTypeCode49	Code 39
BarcodeTypeCode93	Code 49
BarcodeTypeComposite	Code 93
BarcodeTypeD2of5	EAN-UCC Composite Code
BarcodeTypeDataMatrix	Discreate 2 of 5
BarcodeTypeEAN128	Data Matrix
BarcodeTypeEAN13	UCC/EAN-128
BarcodeTypeEAN13CouponCode	EAN-13
BarcodeTypeEAN8	EAN-13 with Extended Coupon Code
BarcodeTypeI2of5	EAN-8
BarcodeTypeIATA	Interleaved 2 of 5
BarcodeTypeISBT128	IATA 2 of 5
BarcodeTypeISBT128Concat	ISBT 128
BarcodeTypeJapanesePost	ISBT-128 Concat.
BarcodeTypeKixPost	Japanese Post
BarcodeTypeKoreaPost	Kix (Netherlands
BarcodeTypeMacroMicroPDF	Korea Post
BarcodeTypeMaxiCode	Macro Micro PDF
BarcodeTypeMicroPDF	MaxiCode
BarcodeTypeMicroQR	Micro PDF 417
BarcodeTypeMSI	MSI
BarcodeTypeMultipacketFormat	OCR
BarcodeTypeOCR	Parameter (FNC3
BarcodeTypeParameterFNC3	PDF-417
BarcodeTypePDF417	Planet Code
BarcodeTypePlanetCode	Plessey Code
BarcodeTypePlesseyCode	PosiCode
BarcodeTypePosiCode	Postnet

BarcodeTypePostnet	QR Code
BarcodeTypeQRCode	Micro QR
BarcodeTypeR2of5	Straight 2 of 5
BarcodeTypeRSS	RSS
BarcodeTypeScanletWebcode	Scanlet Webcode
BarcodeTypeTelepen	Telepen
BarcodeTypeTLC39	TCIF Linked Code 39
BarcodeTypeTriopticCode	Trioptic Code 39
BarcodeTypeUPCA	UPC-A
BarcodeTypeUPCE	UPC-E
BarcodeTypeVeriCode	VeriCode
BarcodeTypeX2of5	Matrix 2 of 5
BarcodeTypeChineseSensible	Chinese-Sensible Code
BarcodeTypeMatrix2of5	Matrix 2 of 5
BarcodeTypeIntelligentMail	Intelligent Mail
BarcodeTypeISSN	ISSN

11.10. ParamName

11.10.1. Zebra(SE4710/SE965)

Flag	Description
ParamNameAimBrightness	Aim Brightness
ParamNameAustraliaPost	Australia Post
ParamNameAustraliaPostFormat	Australia Post Format
ParamNameAztec	Aztec
ParamNameAztecInverse	Aztec Inverse
ParamNameBaudRate	Baud Rate
ParamNameBeepAfterGoodDecode	Beep After Good Decode
ParamNameBeeperDuration	Beeper Duration
ParamNameBeeperTone	Beeper Tone
ParamNameBeeperVolume	Beeper Volume

ParamNameBitsPerPixel	Bits per Pixel (BPP)
ParamNameBooklandEAN	Bookland EAN
ParamNameBooklandISBNFormat	Bookland ISBN Format
ParamNameBootUpEvent	Boot Up Event
ParamNameBufferCode39	Buffer Code 39
ParamNameC2of5	Chinese 2 of 5
ParamNameCheckISBTTable	Check ISBT Table
ParamNameCheckParity	Check Parity
ParamNameCLSIEditing	CLSI Editing
ParamNameCodabar	Codabar
ParamNameCodabarCharDetection	Codabar Upper or Lower Case Start/Stop Characters Detection
ParamNameCodabarLength1	Length for Codabar 1
ParamNameCodabarLength2	Length for Codabar 2
ParamNameCode11	Code 11
ParamNameCode11CheckDigitVerification	Code 11 Check Digit Verification
ParamNameCode11Length1	Length for Code 11 1
ParamNameCode11Length2	Length for Code 11 2
ParamNameCode128	Code 128
ParamNameCode128Emul	Code 128 Emulation
ParamNameCode128Length1	Length for Code 128 1
ParamNameCode128Length2	Length for Code 128 2
ParamNameCode32Prefix	CoCode 32 Prefixde 32 Prefix
ParamNameCode39	Code 39
ParamNameCode39CheckDigitVerification	Code 39 Check Digit Verification
ParamNameCode39FullASCIIConversion	Code 39 Full ASCII Conversion
ParamNameCode39Length1	Length for Code 39 1
ParamNameCode39Length2	Length for Code 39 2
ParamNameCode93	Code 93
ParamNameCode93Length1	Length for Code 93 1
ParamNameCode93Length2	Length for Code 93 2

ParamNameCompositeBeepMode	Composite Beep Mode
ParamNameCompositeCCAB	Composite CC-A/B
ParamNameCompositeCCC	Composite CC-C
ParamNameCompositeTLC39	Composite TLC-39
ParamNameContinuousBarCodeRead	Continuous Bar Code Read
ParamNameConvertCode39toCode32	Convert Code 39 to Code 32
ParamNameConvertGS1DatabarToUPCEAN	Convert GS1 DataBar to UPC/EAN
ParamNameConvertI2of5toEAN13	Convert I 2 of 5 to EAN 13
ParamNameI2of5SecurityLevel	I 2 of 5 Security Level
ParamNameConvertUPCE1toA	Convert UPC-E1 to A
ParamNameConvertUPCEtoA	Convert UPC-E to A
ParamNameCouponReport	Coupon Report
ParamNameBottomPixel	Crop to Pixel Addresses(Bottom
ParamNameLeftPixel	Crop to Pixel Addresses(Left
ParamNameRightPixel	Crop to Pixel Addresses(Right
ParamNameTopPixel	Crop to Pixel Addresses(Top
ParamNameD2of5	Discrete 2 of 5
ParamNameD2of5Length1	Length for D 2 of 5 1
ParamNameD2of5Length2	Length for D 2 of 5 2
ParamNameDataMatrix	Data Matrix
ParamNameDataMatrixInverse	Data Matrix Inverse
ParamNameDecodeAimingPattern	Decode Aiming Pattern
ParamNameDecodeDataPacketFormat	Decode Data Packet Format
ParamNameDecodeEvent	Decode Event
ParamNameDecodeLEDBehavior	Decode LED Behavior
ParamNameDecodeMirrorImages	Decode Mirror Images
ParamNameDecodeSessionTimeout	Decode Session Timeout
ParamNameDecodeUPCEANSupply	Decode UPC/EAN/JAN Supplementals
ParamNameDecodeUPCEANSupplyAIMID	Decode UPC/EAN/JAN Supplemental AIM ID

ParamNameDecodingAutoexposure	Decoding Autoexposure
ParamNameDecodingIllumination	Decoding Illumination
ParamNameEAN13	EAN-13/JAN 13
ParamNameEAN8	EAN-8/JAN 8
ParamNameEAN8Extend	EAN-8/JAN-8 Extend
ParamNameEscapedChar	Escape Characters
ParamNameExposureTime	Exposure Time
ParamNameFixedGain	Fixed Gain
ParamNameFrameRate	Frame Rate
ParamNameFuzzy1DProcessing	Fuzzy 1D Processing
ParamNameGS1128	GS1-128
ParamNameGS1128EmulMode	GS1-128 Emulation Mode for UCC/EAN Composite Codes
ParamNameGS1Databar	GS1 DataBar
ParamNameGS1DatabarExpanded	GS1 DataBar Expanded
ParamNameGS1DatabarLimited	GS1 DataBar Limited
ParamNameGS1DatabarLimitedSecurityLevel	GS1 DataBar Limited Security Level
ParamNameHostCharacterTimeout	Host Character Time-out
ParamNameHostRTSLineState	Host RTS Line State
ParamNameHostSerialResponseTimeout	Host Serial Response Time-out
ParamNameI2of5	Interleaved 2 of 5 (ITF)
ParamNameI2of5CheckDigitVerification	I 2 of 5 Check Digit Verification
ParamNameI2of5Length1	Length for I 2 of 5 1
ParamNameI2of5Length2	Length for I 2 of 5 2
ParamNameImageBrightness	Image Brightness (Target White)
ParamNameImageCaptureAutoexposure	Image Capture Autoexposure
ParamNameImageCaptureIllumination	Image Capture Illumination
ParamNameImageContrastEnhancement	Image Contrast Enhancement
ParamNameImageCropping	Image Cropping
ParamNameImageEdgeSharpening	Image Edge Sharpening

ParamNameImageEnhancement	Image Enhancement
ParamNameImageFileFormatSelection	Image File Format Selection
ParamNameImageFileMetaData	Image File Meta Data
ParamNameImageResolution	Image Resolution
ParamNameImageRotation	Image Rotation
ParamNameIntercharacterGapSize	Intercharacter Gap Size
ParamNameInterpacketDelay	Interpacket Delay
ParamNameInverse1D	Inverse 1D
ParamNameIlluminationBrightness	Illumination Brightness
ParamNameISBT128	ISBT 128
ParamNameISBTConcatenation	ISBT Concatenation
ParamNameISBTConcatenationRedundancy	ISBT Concatenation Redundancy
ParamNameISSNEAN	ISSN EAN
ParamNameJapanPostal	Japan Postal
ParamNameJPEGImageOptions	JPEG Image Options
ParamNameJPEGQualityValue	JPEG Quality Value
ParamNameJPEGSizeValue	JPEG Size Value
ParamNameK3of5	Korean 3 of 5
ParamNameLEDIllumination	LED Illumination
ParamNameLockParameterScanning	Lock Parameter Scanning
ParamNameLowLightMotionDetection	Low Light Motion Detection
ParamNameM2of5	Matrix 2 of 5
ParamNameM2of5CheckDigit	Matrix 2 of 5 Check Digit
ParamNameM2of5Length1	Length for Matrix 2 of 5 1
ParamNameM2of5Length2	Length for Matrix 2 of 5 2
ParamNameMacroPDFTransmit	Macro PDF Transmit/Decode Mode Symbols
ParamNameMaxicode	Maxicode
ParamNameMicroPDF417	MicroPDF417
ParamNameMicroQR	MicroQR
ParamNameMirroredImage	Mirrored Image

ParamNameMobilePhoneDisplayMode	Mobile Phone/Display Mode
ParamNameMSI	MSI
ParamNameMSICheckDigit	MSI Check Digits
ParamNameMSICheckDigitAlgorithm	MSI Check Digit Algorithm
ParamNameMSILength1	Length for MSI 1
ParamNameMSILength2	Length for MSI 2
ParamNameMultipacketOption	Multipacket Option
ParamNameNetherlandsKIXCode	Netherlands KIX Code
ParamNameNOTISEditing	NOTIS Editing
ParamNameParameterEvent	Parameter Event
ParamNameParameterScanning	Parameter Scanning
ParamNameParity	Parity
ParamNamePDF417	PDF417
ParamNamePDFPrioritization	PDF Prioritization
ParamNamePDFPrioritizationTimeout	PDF Prioritization Timeout
ParamNamePicklistMode	Picklist Mode
ParamNamePowerMode	Power Mode
ParamNamePresentationModeFieldofView	Presentation Mode Field of View
ParamNameQRCode	QR Code
ParamNameQRInverse	QR Inverse
ParamNameRedundancyLevel	Redundancy Level
ParamNameScanDataTransmissionFormat	Scan Data Transmission Format
ParamNameSecurityLevel	Security Level
ParamNameSnapshotAimingPattern	Snapshot Aiming Pattern
ParamNameSnapshotModeTimeout	Snapshot Mode Timeout
ParamNameSoftwareHandshaking	Software Handshaking
ParamNameSSIPrefix	SSI Prefix Value
ParamNameSSISuffix1	SSI Suffix 1 Value
ParamNameSSISuffix2	SSI Suffix 2 Value
ParamNameStopBits	Stop Bits
ParamNameSuppressPowerupBeeps	Suppress Power-up Beeps

ParamNameTargetVideoFrameSize	Target Video Frame Size
ParamNameTimeDelaytoLowPowerMode	Time Delay to Low Power Mode
ParamNameTimeoutBetweenDecodesDifferentSymbols	Timeout Between Decodes, Different Symbols
ParamNameTimeoutBetweenDecodesSameSymbol	Timeout Between Decodes, Same Symbol
ParamNameTransmitCode11CheckDigit	Transmit Code 11 Check Digit
ParamNameTransmitCode39CheckDigit	Transmit Code 39 Check Digit
ParamNameTransmitCodeIDCharacter	Transmit Code ID Character
ParamNameTransmitI2of5CheckDigit	Transmit I 2 of 5 Check Digit
ParamNameTransmitM2of5CheckDigit	Transmit Matrix 2 of 5 Check Digit
ParamNameTransmitMacroPDFControlHeader	Transmit Macro PDF Control Header
ParamNameTransmitMSICheckDigit	Transmit MSI Check Digit
ParamNameTransmitNoRead	Transmit "No Read" Message
ParamNameTransmitUKPostalCheckDigit	Transmit UK Postal Check Digit
ParamNameTransmitUPCACheckDigit	Transmit UPC-A Check Digit
ParamNameTransmitUPCE1CheckDigit	Transmit UPC-E1 Check Digit
ParamNameTransmitUPCECheckDigit	Transmit UPC-E Check Digit
ParamNameTransmitUSPostalCheckDigit	Transmit US Postal Check Digit
ParamNameTriggerModes	Trigger Modes
ParamNameTriopticCode39	Trioptic Code 39
ParamNameUCCCouponExtendCode	UCC Coupon Extended Code
ParamNameUKPostal	UK Postal
ParamNameUniqueBarCodeReporting	Unique Bar Code Reporting
ParamNameUnlockParameterScanning	Unlock Parameter Scanning
ParamNameUPCA	UPC-A
ParamNameUPCAPreamble	UPC-A Preamble
ParamNameUPCCompositeMode	UPC Composite Mode
ParamNameUPCE	UPC-E
ParamNameUPCE1	UPC-E1

ParamNameUPCE1Preamble	UPC-E1 Preamble
ParamNameUPCEANSupplyRedundancy	UPC/EAN/JAN Supplemental Redundancy
ParamNameUPCEPreamble	UPC-E Preamble
ParamNameUPUFICSPostal	UPU FICS Postal
ParamNameUserParameterPassThrough	User Parameter Pass Through
ParamNameUserProgrammableSupply1	User-Programmable Supplementals 1
ParamNameUserProgrammableSupply2	User-Programmable Supplementals 2
ParamNameUSPlanet	US Planet
ParamNameUSPostnet	US Postnet
ParamNameUSPS4CB	USPS 4CB/One Code/Intelligent Mail
ParamNameValidateConcatenatedParameterBarCodes	Validate Concatenated Parameter Bar Codes
ParamNameVideoViewFinder	Video View Finder
ParamNameVideoViewFinderImageSize	Video View Finder Image Size
ParamNameVideoResolution	Video Resolution
ParamNameVisualDecodeIndicatorDecodeBlinkDuration	Visual Decode Indicator Decode Blink Duration
ParamNameVisualDecodeIndicatorDecodeBlinks	Visual Decode Indicator Decode Blinks
ParamNamecode128ReducedQuietZone	Code128 Reduced Quiet Zone
ParamNameIgnoreCode128	Ignore Code 128 <FNC4>
ParamNameUPCReducedQuietZone	UPC Reduced Quiet Zone
ParamNameCode39QuietZone	Code39 Reduced Quiet Zone
ParamNameI2of5QuietZone	I2 of 5 Reduced Quiet Zone
ParamNameHanXin	Chinese Sensible(Han Xin)
ParamNameHanXinInverse	Chinese Sensible Inverse

11.10.2. Honeywell(N3680/N6603)

Flag	Description
------	-------------

Codabar	Codabar
CodabarStartStopChar	Codabar Start/Stop Character
CodabarCheckChar	Codabar Check Char
CodabarConcatenation	Codabar Concatenation
CodabarLengthMin	Codabar Message Length Min
CodabarLengthMax	Codabar Message Length Max
Code39	Code 39
Code39StartStopChar	Code 39 Start/Stop Character
Code39CheckChar	Code 39 Check Char
Code39LengthMin	Code 39 Message Length Min
Code39LengthMax	Code 39 Message Length Max
Code39Append	Code 39 Append
Code39Pharmaceutical	Code 32 Pharmaceutical
Code39FullAscii	Code 39 Full ASCII
Code39CodePage	Code 39 Code Page
I2of5	Interleaved 2 of 5
I2of5CheckDigit	Interleaved 2 of 5 Check Digit
I2of5LengthMin	Interleaved 2 of 5 Message Length Min
I2of5LengthMax	Interleaved 2 of 5 Message Length Max
NEC2of5	NEC 2 of 5
NEC2of5CheckDigit	NEC 2 of 5 Check Digit
NEC2of5LengthMin	NEC 2 of 5 Message Length Min
NEC2of5LengthMax	NEC 2 of 5 Message Length Max
Code93	Code 93
Code93LengthMin	Code 93 Message Length Min
Code93LengthMax	Code 93 Message Length Max
Code93Append	Code 93 Append
Code93CodePage	Code 93 Code Page
R2of5	Straight 2 of 5 Industrial

R2of5LengthMin	Straight 2 of 5 Industrial Message Length Min
R2of5LengthMax	Straight 2 of 5 Industrial Message Length Max
A2of5	Straight 2 of 5 IATA
A2of5LengthMin	Straight 2 of 5 IATA Message Length Min
A2of5LengthMax	Straight 2 of 5 IATA Message Length Max
X2of5	Matrix 2 of 5
X2of5LengthMin	Matrix 2 of 5 Message Length Min
X2of5LengthMax	Matrix 2 of 5 Message Length Max
Code11	Code 11
Code11CheckDigit	Code 11 Check Digit
Code11LengthMin	Code 11 Message Length Min
Code11LengthMax	Code 11 Message Length Max
Code128	Code 128
IsbtConcatenation	ISBT Concatenation
Code128LengthMin	Code 128 Message Length Min
Code128LengthMax	Code 128 Message Length Max
Code128Append	Code 128 Append
Code128CodePage	Code 128 Code Page
GS1128	GS1-128
GS1128LengthMin	GS1-128 Message Length Min
GS1128LengthMax	GS1-128 Message Length Max
UPCA	UPC-A
UPCACheckDigit	UPC-A Check Digit
UPCANumberSystem	UPC-A Number System
UPCA2DigitAdd	UPC-A 2 Digit Addenda
UPCA5DigitAdd	UPC-A 5 Digit Addenda
UPCAAddReq	UPC-A Addenda Required
UPCAAddSep	UPC-A Addenda Separator

UPCACouponCode	UPC-A/EAN-13 with Extended Coupon Code
CouponGS1DataBarOutput	Coupon GS1 DataBar Output
ConvertUPCAtoEAN13	Convert UPC-A to EAN-13
UPCE0	UPC-E0
UPCE0Expand	UPC-E0 Expand
UPCE0CheckDigit	UPC-E0 Check Digit
UPCE0NumberSystem	UPC-E0 Number System
UPCE02DigitAdd	UPC-E0 2 Digit Addenda
UPCE05DigitAdd	UPC-E0 5 Digit Addenda
UPCE0AddReq	UPC-E0 Addenda Required
UPCE0AddSep	UPC-E0 Addenda Separator
UPCE1	UPC-E1
EAN13	EAN/JAP-13
EAN13CheckDigit	EAN/JAP-13 Check Digit
EAN132DigitAdd	EAN/JAP-13 2 Digit Addenda
EAN135DigitAdd	EAN/JAP-13 5 Digit Addenda
EAN13AddReq	EAN/JAP-13 Addenda Required
EAN13AddSep	EAN/JAP-13 Addenda Separator
IsbnTranslate	ISBN Translate
EAN8	EAN/JAP-8
EAN8CheckDigit	EAN/JAP-8 Check Digit
EAN82DigitAdd	EAN/JAP-8 2 Digit Addenda
EAN85DigitAdd	EAN/JAP-8 5 Digit Addenda
EAN8AddReq	EAN/JAP-8 Addenda Required
EAN8AddSep	EAN/JAP-8 Addenda Separator
MSI	MSI
MSICheckChar	MSI Check Character
MSILengthMin	MSI Message Length Min
MSILengthMax	MSI Message Length Max
RSS14	RSS-14

RSSLimit	RSS Limited
RSSExp	RSS Expanded
RSSExpLengthMin	RSS Expanded Length Min
RSSExpLengthMax	RSS Expanded Length Max
CodablockA	Codablock A
CodablockALengthMin	Codablock A Length Min
CodablockALengthMax	Codablock A Length Max
CodablockF	Codablock F
CodablockFLengthMin	Codablock F Length Min
CodablockFLengthMax	Codablock F Length Max
PDF417	PDF 417
PDF417LengthMin	PDF 417 Message Length Min
PDF417LengthMax	PDF 417 Message Length Max
MacroPDF417	MacroPDF417
MicroPDF	MicroPDF 417
MicroPDFLengthMin	MicroPDF 417 Length Min
MicroPDFLengthMax	MicroPDF 417 Length Max
ComCode	EAN/UCC Composite Code
UPCEANVersion	UPC/EAN Version
ComCodeLengthMin	EAN/UCC Composite Code Message Length Min
ComCodeLengthMax	EAN/UCC Composite Code Message Length Max
GS1Emulation	GS1 Emulation
TLC39	TCIF Linked Code 39
ChinaPost	China Post
ChinaPostLengthMin	China Post Message Length Min
ChinaPostLengthMax	China Post Message Length Max
KoreaPost	Korea Post
KoreaPostLengthMin	Korea Post Message Length Min
KoreaPostLengthMax	Korea Post Message Length Max

KoreaPostCheckDigit	Korea Post Check Digit
QRCode	QR Code
QRCodeLengthMin	QR Code Message Length Min
QRCodeLengthMax	QR Code Message Length Max
QRCodeAppend	QR Code Append
QRCodePage	QR Code Page
Matrix	Data Matrix
MatrixLengthMin	Data Matrix Message Length Min
MatrixLengthMax	Data Matrix Message Length Max
MatrixAppend	Data Matrix Append
MatrixCodePage	Data Matrix Code Page
MaxiCode	MaxiCode
MaxiCodeLengthMin	MaxiCode Message Length Min
MaxiCodeLengthMax	MaxiCode Message Length Max
AztecCodeDefault	Default All Aztec Code Settings
AztecCode	Aztec Code
AztecCodeLengthMin	Aztec Code Message Length Min
AztecCodeLengthMax	Aztec Code Message Length Max
AztecAppend	Aztec Append
AztecCodePage	Aztec Code Page
HanXinCode	Chinese Sensible (Han Xin) Code
HanXinCodeLengthMin	Chinese Sensible (Han Xin) Code Message Length Min
HanXinCodeLengthMax	Chinese Sensible (Han Xin) Code Message Length Max
PostalCodes	2D Postal Codes
PlanetCodeCheckDigit	Planet Code Check Digit
PostnetCheckDigit	Postnet Check Digit
AustralianPostInterpretation	Australian Post Interpretation