



ASR-P252B

C# SDK リファレンスガイド

Revision History

バージョン	修正内容	改訂履歴	日付
1.0	初期バージョン	トリニティ・リウ	2022/12/01

内容

はじめに.....	9
1. 開発環境の構築.....	10
1.1. SDK の追加.....	10
1.1.1. Windows デスクトップアプリケーションを作成する.....	10
1.1.2. リファレンスを追加.....	10
1.2. 名前空間を参照する.....	13
1.3. SDK の使用方法.....	13
1.3.1. 機器との接続.....	13
1.3.2. タグをインベントリ.....	14
1.3.3. AsReader との接続を解除する.....	14
2. AsReader クラス.....	15
2.1. 関数.....	15
2.1.1. Connect.....	15
2.1.2. Disconnect.....	15
2.1.3. SetDelegates.....	16
2.1.4. GetIdleTime.....	16
2.1.5. Connect.....	17
2.1.6. GetLinkProfile.....	18
2.1.7. GetBaudRate.....	18
2.1.8. GetRssiOnOff.....	19
2.1.9. GetHardVersion.....	19

2.1.10.	GetQMinValue	20
2.1.11.	GetQValue	21
2.1.12.	GetSerialNumber	21
2.1.13.	GetRFIDModuleVersion	22
2.1.14.	GetQMaxValue	22
2.1.15.	GetPowerRange.....	23
2.1.16.	GetBuzzer.....	24
2.1.17.	GetBattery	24
2.1.18.	GetSession	25
2.1.19.	GetSessionFlag.....	25
2.1.20.	GetRegionCountry	26
2.1.21.	GetOperationTime	27
2.1.22.	GetInventoryMode.....	27
2.1.23.	GetPower.....	28
2.1.24.	GetAutoOffTime	28
2.1.25.	GetFirmwareVersion.....	29
2.1.26.	GetAutoOffSleep	30
2.1.27.	GetBaudRateList	30
2.1.28.	GetBarcodeTimeout.....	31
2.1.29.	GetFrequencyList.....	31
2.1.30.	GetFrequency	32
2.1.31.	RestoreFactorySettings.....	33
2.1.32.	SetBuzzer	33
2.1.33.	SetIdleTime.....	34
2.1.34.	SetAutoOffSleep.....	34

2.1.35.	SetAutoOffTime.....	35
2.1.36.	SetQValue	36
2.1.37.	SetBarcodeTimeout.....	36
2.1.38.	SetLinkProfile.....	37
2.1.39.	SetFrequencyAutomatic.....	38
2.1.40.	SetSession	38
2.1.41.	SetInventoryMode	39
2.1.42.	SetSessionFlag	40
2.1.43.	SetPowerGain	40
2.1.44.	SetRssiOnOff.....	41
2.1.45.	SetFrequencyAssigned.....	42
2.1.46.	SetBaudRate.....	42
2.1.47.	SetOperationTime.....	43
2.1.48.	Kill	44
2.1.49.	StartBarcodeScan	44
2.1.50.	StartInventory.....	45
2.1.51.	StartInventoryMoisture.....	46
2.1.52.	StartInventoryTemperature	46
2.1.53.	Stop.....	47
2.1.54.	GetEPCMaskData	48
2.1.55.	SetEPCMaskData.....	48
2.1.56.	ClearEPCMaskData	49
2.1.57.	GetTemperature.....	49
2.1.58.	Read.....	50
2.1.59.	Write	51

2.1.60.	Lock.....	52
2.1.61.	PermaLock	53
2.1.62.	UnLock.....	54
2.1.63.	SetAccessPassword	55
2.1.64.	SetKillPassword	56
3.	AsReaderDelegate クラス.....	57
3.1.	デリゲート.....	57
3.1.1.	WhenReceivedFirmwareVersion	57
3.1.2.	WhenReceivedHardVersion	57
3.1.3.	WhenReceivedRegionCountry.....	58
3.1.4.	WhenReceivedSerialNumber.....	58
3.1.5.	WhenReceivedRFIDModuleVersion	59
3.1.6.	WhenReceivedBuzzer.....	60
3.1.7.	WhenReceivedAutoOffSleep.....	60
3.1.8.	WhenReceivedAutoOffTime	61
3.1.9.	WhenReceivedQValue	61
3.1.10.	WhenReceivedQMaxValue	62
3.1.11.	WhenReceivedQMinValue	62
3.1.12.	WhenReceivedTemperature	63
3.1.13.	WhenReceivedIdleTime	64
3.1.14.	WhenReceivedBattery.....	64
3.1.15.	WhenReceivedBarcodeTimeout.....	65
3.1.16.	WhenReceivedRestoreFactorySettings.....	65
3.1.17.	WhenReceivedLinkProfile	66
3.1.18.	WhenReceivedBaudRate.....	66

3.1.19.	WhenReceivedBaudRateList	67
3.1.20.	WhenReceivedPowerRange.....	68
3.1.21.	WhenReceivedFrequencyList	68
3.1.22.	WhenReceivedFrequency	69
3.1.23.	WhenReceivedSetFrequencyResult.....	69
3.1.24.	WhenReceivedSetBuzzerResult	70
3.1.25.	WhenReceivedSetQValueResult.....	71
3.1.26.	WhenReceivedSetIdleTimeResult.....	71
3.1.27.	WhenReceivedSetLinkProfileResult.....	72
3.1.28.	WhenReceivedSetAutoOffSleepResult.....	72
3.1.29.	WhenReceivedSetAutoOffTimeResult.....	73
3.1.30.	WhenReceivedSetBarcodeTimeoutResult	74
3.1.31.	WhenReceivedStartBarcodeScanResult	74
3.1.32.	WhenReceivedStopResult.....	75
3.1.33.	WhenReceivedBarcodeData	75
3.1.34.	WhenReceivedReadTagData	76
3.1.35.	WhenReceivedData	77
3.1.36.	WhenReceivedTagData	78
3.1.37.	WhenReceivedEPCMaskData.....	78
3.1.38.	WhenReceivedClearEPCMaskResult.....	79
3.1.39.	WhenReceivedSetEPCMakResult	79
3.1.40.	WhenReceivedWorkTemperature	80
3.1.41.	WhenReceivedSetSessionResult.....	81
3.1.42.	WhenReceivedSetBaudRateResult	81
3.1.43.	WhenReceivedSetSessionFlagResult	82

3.1.44.	WhenReceivedSetInventoryModeResult	83
3.1.45.	WhenReceivedSetPowerGainResult	83
3.1.46.	WhenReceivedSetOperationTimeResult	84
3.1.47.	WhenReceivedSetInventoryTimeResult	84
3.1.48.	WhenReceivedSetRssiOnOffResult	85
3.1.49.	WhenReceivedReadResult	86
3.1.50.	WhenReceivedWriteResult	86
3.1.51.	WhenReceivedLockResult	87
3.1.52.	WhenReceivedPermaLockResult	87
3.1.53.	WhenReceivedKillResult	88
3.1.54.	WhenReceivedOperationTime	88
3.1.55.	WhenReceivedInventoryMode	89
3.1.56.	WhenReceivedRssiOnOff	90
3.1.57.	WhenReceivedPowerGain	90
3.1.58.	WhenReceivedSession	91
3.1.59.	WhenReceivedSessionFlag	91
3.1.60.	WhenReceivedStartInventoryResult	92
3.1.61.	WhenReceivedTriggerStatus	92
3.1.62.	WhenReceivedModeStatus	93
3.1.63.	WhenDeviceIsConnected	94
4.	タイプクラス	95
4.1.	列挙型	95
4.1.1.	BuzzerStatus	95
4.1.2.	InventoryMode	95
4.1.3.	RssiOnOff	95

4.1.4.	Baudrate.....	96
4.1.5.	DataArea	96
4.1.6.	ResultCode.....	96
4.1.7.	CommandType	97
4.1.8.	TagType	97
4.1.9.	ButtonActionStatus.....	97
4.1.10.	BluetoothType	98
5.	構造物	99
5.1.	TagData	99
6.	AsBluetoothManager クラス	100
6.1.	関数	100
6.1.1.	StartSearching	100
6.1.2.	StopSearching	100
6.2.	デリゲート.....	101
6.2.1.	WhenReceivedDeviceList.....	101
7.	BluetoothDevice クラス.....	102
7.1.	プロパティ.....	102

はじめに

本書は、SDKを使用して Windows デスクトップアプリケーションを開発する開発者向けに、以下の情報を提供するものです。

- 開発環境の構築方法
- 各種 SDK ライブラリ関数の説明

開発ツール。

- Visual Studio 2019

1. 開発環境の構築

1.1. SDK の追加

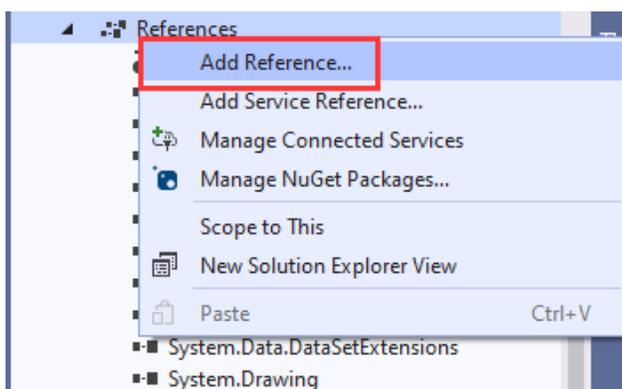
1.1.1. Windows デスクトップアプリケーションを作成する

SDK をプロジェクトフォルダーにコピーします。

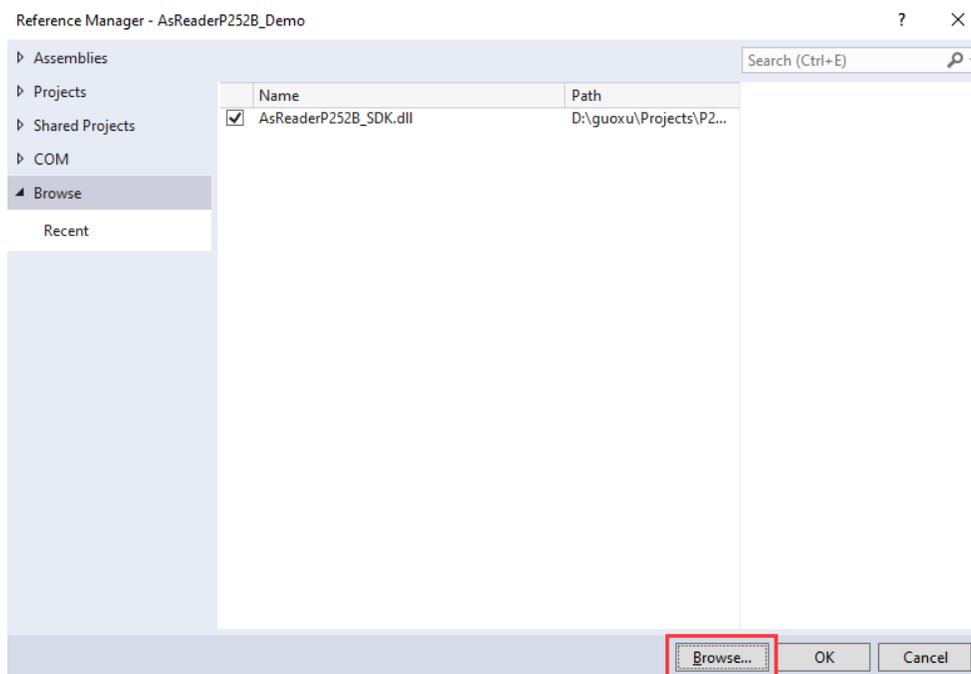
開発またはデバッグの場合は、プログラムのカレントディレクトリの bin/Debug フォルダにコピーしてください。リリースする場合は、プログラムのカレントディレクトリの bin/Release フォルダにコピーしてください。

1.1.2. リファレンスを追加

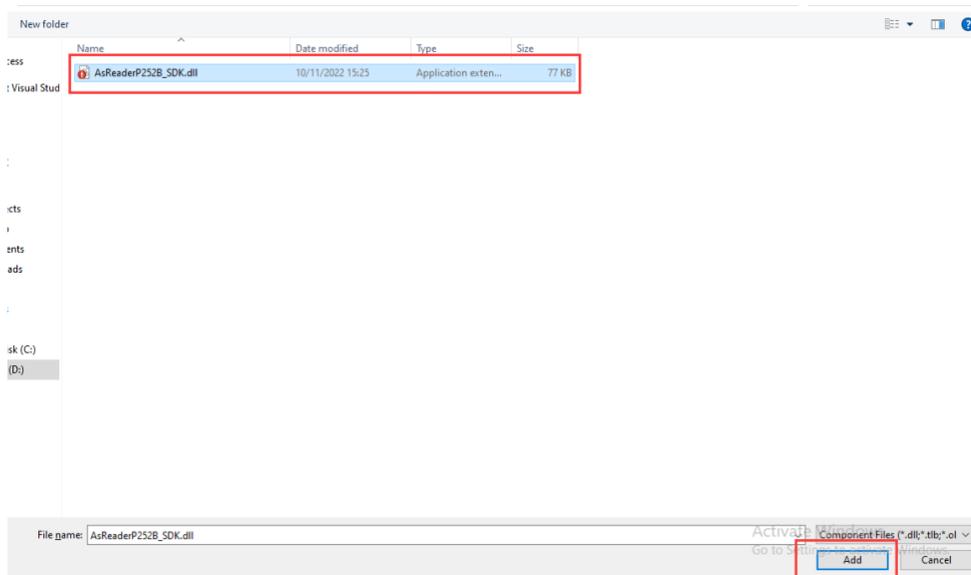
「References」を右クリックし、「Add Reference」を選択します。



「Browse」ボタンをクリックします。

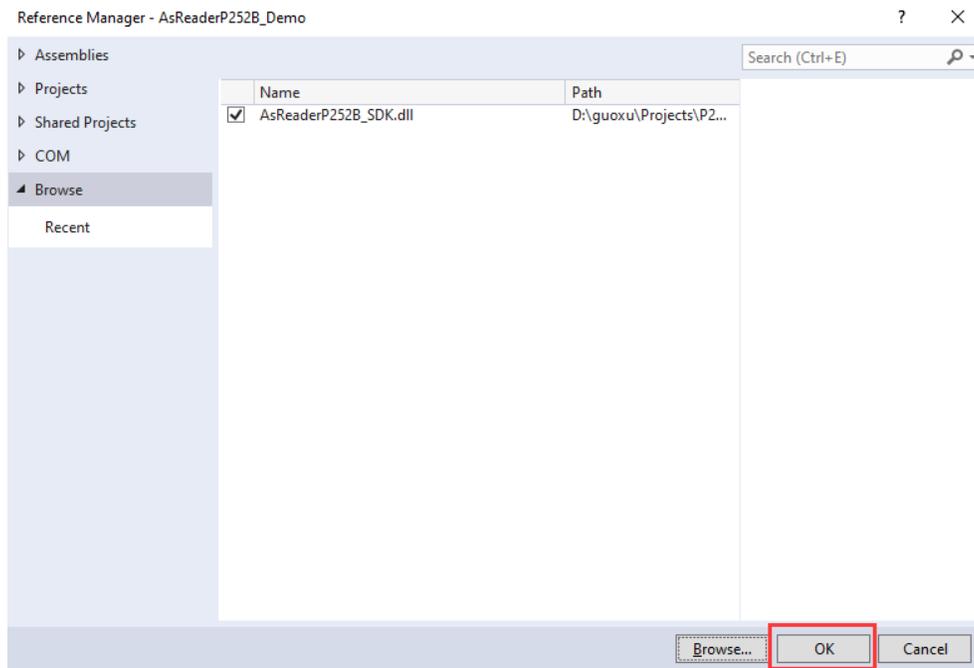


「Select the files reference」ダイアログボックスで、プロジェクトパスの下にある SDK ファイルを選択し、「Add」ボタンをクリックします。

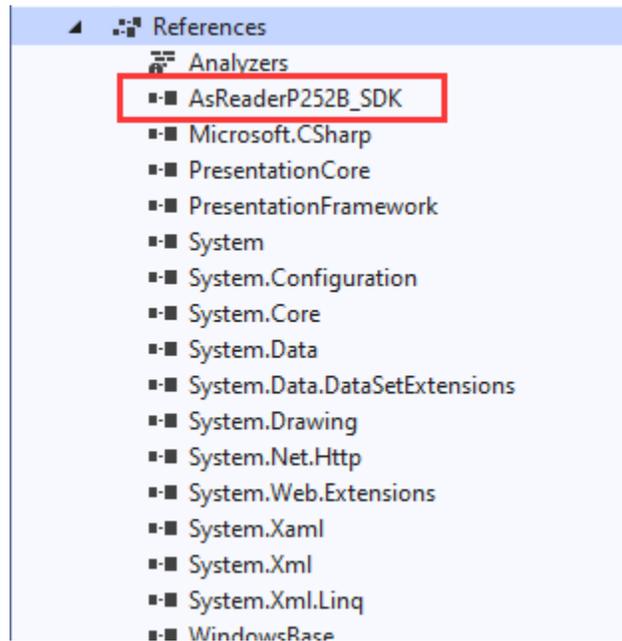


Reference Manager ページで 「OK」ボタンをクリックします。この時点で、SDK ファイルの追加

は完了です。



正常に追加されると、参照リストに表示されます。



1.2. 名前空間を参照する

```
Using AsReaderP252B;
```

1.3. SDK の使用方法

AsReader クラスの関数の呼び出し、AsReader との接続・切断、タグの読み取りを例に説明します。

1.3.1. 機器との接続

1. AsReader オブジェクトの取得

AsReader クラスのコンストラクタを呼び出して、AsReader のオブジェクトを取得します。(以下はサンプルコード。)

```
AsReader asreader = new AsReader();
```

2. AsReaderDelegate オブジェクトの取得

AsReaderDelegate クラスのデフォルトコンストラクタ関数を呼び出し、AsReaderDelegate オブジェクトを取得します。

```
AsReaderDelegate asReaderDelegate = new AsReaderDelegate();
```

3. AsReader と接続

COM ポートとボーレートをパラメータにセットして Connect 関数を呼び出します。接続に成功した場合、この関数は 0 を返します。

```
int ret = asreader.Connect("COM1", 115200);  
if(ret == 0){  
    // 接続成功後のプログラム。  
}else{  
    // 接続に失敗した後のプログラム。  
}
```

1.3.2. タグをインベントリ

1. AsReaderDelegate オブジェクトのデリゲートを指定

(1) TagData 型のパラメータを持つ関数を作成します。

本関数は、AsReader がタグをインベントリする際のコールバック処理に使用されます。パラメータ tagData は TagData 型です (5.1 参照)。

```
private void ReceivedTagData(TagData tagData)
{
    //インベントリした RFID データを処理します。
}
```

(2) WhenReceivedTagData デリゲートを指定

AsReaderDelegate クラスの whenReceivedTagData デリゲートに 1.3.2-1-(1) で作成した関数を指定します。

```
asReaderDelegate. whenReceivedTagData= ReceivedTagData;
```

2. タグをインベントリ

AsReader クラスの関数 public UInt32 StartInventory()を呼び出して、タグのインベントリを行います。タグのインベントリが完了すると、1.3.2-1(1)で作成した関数がコールバックされ、読み取り結果が TagData 型で渡されます。

```
//インベントリータグ
UInt32 resultCode = asreader.StartInventory();
```

1.3.3. AsReader との接続を解除する

AsReader クラスの関数「public UInt32 Disconnect()」を呼び出して、AsReader との接続を解除します。切断されると、この関数は 0 を返します。

```
UInt32 ret = asReader.Disconnect();
if(ret == 0)
{
    // 端末との切断に成功時の処理
}else{
    // 端末との切断に失敗時の処理
}
```

2. AsReader クラス

2.1. 関数

2.1.1. Connect

関数名	UInt32 Connect(string comPort, int baudrate);			
パラメータ	パラメータ名	イン/アウト	型	説明
	comPort	In	string	COM ポート
	baudrate	In	int	ボーレート
戻り値	-	Out	UInt32	関数呼び出しの実行結果 0 : 成功 1 : 失敗
<p>関数の説明 : COM ポートから AsReader に接続します。</p> <p>呼び出し条件 : なし。</p> <p>サンプルコード : // AsReader に接続する (ポート : COM1、ボーレート : 115200bps) 。 UInt32 ret = Device.Connect("COM1", 115200);</p>				

2.1.2. Disconnect

関数名	UInt32 Disconnect();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	Int	関数を呼び出したときの実行結果。 0 : 成功

				1: 失敗
関数の説明： AsReader との接続を解除します。				
呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。				
サンプルコード： <pre>UInt32 ret = Device.Disconnect();</pre>				

2.1.3. SetDelegates

関数名	void SetDelegates(AsReaderDelegate delegates);			
パラメータ	パラメータ名	イン/アウト	型	説明
	delegates	In	AsReaderDelegate	AsReaderDelegate オブジェクト(3 参照)
戻り値	-	-	-	-
関数の説明： AsReaderDelegate オブジェクトにデリゲート関数を設定します。				
呼び出し条件： なし。				
呼び出し条件： Device.SetDelegates(delegates)を実行します。 // delegates は AsReaderDelegate クラスのインスタンスオブジェクト。				

2.1.4. GetIdleTime

関数名	UInt32 GetIdleTime();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-

戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のアイドルタイムを取得する。 この関数が呼ばれると、その結果は"WhenReceivedIdleTime" (3.1.13 参照) を介して、AsReader のアイドルタイムを返します。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetIdleTime();</p>				

2.1.5. Connect

関数名	UInt32 Connect(BluetoothDevice bluetoothDevice);			
パラメータ	パラメータ名	イン/アウト	型	説明
	bluetoothDevice	In	BluetoothDevice	接続されている Bluetooth 機器オブジェクト BluetoothDevice タイプ (7参照)
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : Bluetooth で AsReader と接続します。 この関数が呼ばれると、AsReader との接続状態がデリゲート "WhenDeviceIsConnected" (3.1.63 参照) を介して返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード :</p>				

```
Device.Connect(device);
```

2.1.6. GetLinkProfile

関数名	UInt32 GetLinkProfile();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のリンクプロフィールを取得します。 この関数が呼ばれると、"WhenReceivedLinkProfile" (3.1.17 参照) を介して、AsReader のリンクプロフィールが結果として返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetLinkProfile()。</p>				

2.1.7. GetBaudRate

関数名	UInt32 GetBaudRate();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の Serial ポート通信ボーレート値を取得します。</p>				

この関数が呼ばれると、"WhenReceivedBaudRate" (3.1.18 参照) を通じて AsReader の Serial ポート通信ボーレート値が結果として返されます。

注意：この関数を呼び出す前に、関数 GetBaudRateList (2.1.17 参照) を呼び出して AsReader で利用可能なボーレートのリストを取得する必要があります。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetBaudRate();
```

2.1.8. GetRssiOnOff

関数名	UInt32 GetRssiOnOff();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗

関数の説明：

AsReader の RSSI On/Off 状態を取得します。

この関数が呼ばれると、"WhenReceivedRssiOnOff" (3.1.56 参照) を介して、AsReader の RSSI On/Off 状態が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetRssiOnOff();
```

2.1.9. GetHardVersion

関数名	UInt32 GetHardVersion();
-----	--------------------------

パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明 :
AsReader のハードウェアバージョンを取得します。
この関数が呼ばれると、その結果は"WhenReceivedHardVersion" (3.1.2 参照) を介して、AsReader のハードウェアバージョンを返します。

呼び出し条件 :
USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード :
Device.GetHardVersion();

2.1.10. GetQMinValue

関数名	UInt32 GetQMinValue();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明 :
AsReader の最小 Q を取得します。
この関数が呼ばれると、その結果は"WhenReceivedQMinValue" (3.1.11 参照) を介して、AsReader の最小 Q が返されます。

呼び出し条件 :
USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetQMinValue();
```

2.1.11. GetQValue

関数名	UInt32 GetQValue();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明：

AsReader の Q 値を取得します。

この関数が呼ばれると、その結果は"WhenReceivedQValue" (3.1.9 参照) を介して AsReader の Q 値が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetQValue();
```

2.1.12. GetSerialNumber

関数名	UInt32 GetSerialNumber();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明：

AsReader のシリアル番号を取得します。

この関数が呼ばれると、AsReaderDelegate クラスの "WhenReceivedSerialNumber" (3.1.4 参照) というデリゲートを介して、AsReader のシリアルナンバーが返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetSerialNumber();
```

2.1.13. GetRFIDModuleVersion

関数名	UInt32 GetRFIDModuleVersion();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗

関数の説明：

AsReader の RFID モジュールバージョンを取得します。

この関数が呼ばれると、"WhenReceivedRFIDModuleVersion" (3.1.5 参照) を介して、AsReader の RFID モジュールバージョンが返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
デバイス GetRFIDModuleVersion();
```

2.1.14. GetQMaxValue

関数名	UInt32 GetQMaxValue();			
パラメータ	パラメータ名	イン/アウト	型	説明

		ト		
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : AsReader の Maxmum Q を取得します。 この関数が呼ばれると、その結果は"WhenReceivedQMaxValue" (3.1.10 参照) を介して、AsReader の Maxmum Q を返します。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetQMaxValue();</p>				

2.1.15. GetPowerRange

関数名	UInt32 GetPowerRange();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : AsReader のアンテナパワーの範囲を取得します。 この関数が呼ばれると、"WhenReceivedPowerRange" (3.1.20 参照) を介して、AsReader のアンテナパワーの範囲が返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p>				

サンプルコード：

```
Device.GetPowerRange();
```

2.1.16. GetBuzzer

関数名	UInt32 GetBuzzer();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
関数の説明： AsReader のブザーの状態を取得します。 この関数が呼ばれると、その結果はデリゲート "WhenReceivedBuzzer" (3.1.6 参照)を介して、AsReader のブザー状態を返します。				
呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。				
サンプルコード： Device.GetBuzzer;				

2.1.17. GetBattery

関数名	UInt32 GetBattery();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
関数の説明：				

AsReader の電池残量を取得します。

この関数が呼ばれると、"WhenReceivedBattery"(3.1.14 参照)を介して、AsReader のバッテリー残量が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetBattery();
```

2.1.18. GetSession

関数名	UInt32 GetSession();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明：

AsReader の Session 状態を取得します。

この関数が呼ばれると、その結果は"WhenReceivedSession" (3.1.58 参照) を介して、AsReader の Session 状態が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetSession();
```

2.1.19. GetSessionFlag

関数名	UInt32 GetSessionFlag();			
パラメータ	パラメータ名	イン/アウト	型	説明

	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のセッションフラグを取得します。 この関数が呼ばれると、その結果は"WhenReceivedSessionFlag" (3.1.59 参照) を介して AsReader の Session Flag が返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetSessionFlag();</p>				

2.1.20. GetRegionCountry

関数名	UInt32 GetRegionCountry();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader が適用される地域または国を取得します。 この関数が呼ばれると、その結果は"WhenReceivedRegionCountry" (3.1.3 参照) を介して、AsReader が該当する地域や国を返します。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード :</p>				

```
Device.GetRegionCountry();
```

2.1.21. GetOperationTime

関数名	UInt32 GetOperationTime();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : AsReader の操作時間を取得します。 この関数が呼ばれると、その結果は "WhenReceivedOperationTime" (3.1.54 参照) を介して AsReader の操作時刻を返します。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetOperationTime();</p>				

2.1.22. GetInventoryMode

関数名	UInt32 GetInventoryMode();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : AsReader のインベントリーモードを取得します。</p>				

この関数が呼ばれると、その結果は"WhenReceivedInventoryMode" (3.1.55 参照) を介して、AsReader のインベントリモードを返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetInventoryMode();
```

2.1.23. GetPower

関数名	UInt32 GetPower();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
<p>関数の説明： AsReader の現在のアンテナパワーを取得します。 この関数が呼ばれると、"WhenReceivedPowerGain" (3.1.57 参照) を介して、AsReader の現在のアンテナパワーを返す結果が返されます。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード： Device.GetPower();</p>				

2.1.24. GetAutoOffTime

関数名	UInt32 GetAutoOffTime();			
パラメータ	パラメータ名	イン/アウト	型	説明

	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のオートオフ時間を取得します。 この関数が呼ばれると、"WhenReceivedAutoOffTime" (3.1.8 参照) を介して、AsReader のオートオフタイムを返す結果が返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetAutoOffTime();</p>				

2.1.25. GetFirmwareVersion

関数名	UInt32 GetFirmwareVersion();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のファームウェアのバージョンを取得します。 この関数が呼ばれると、その結果は"WhenReceivedFirmwareVersion" (3.1.1 参照) を通じて、AsReader のファームウェアバージョンが返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p>				

サンプルコード：

```
Device.GetFirmwareVersion();
```

2.1.26. GetAutoOffSleep

関数名	UInt32 GetAutoOffSleep();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗

関数の説明：

AsReader のスリープ時間を取得します。

この関数が呼ばれると、"WhenReceivedAutoOffSleep" (3.1.7 参照) を介して、AsReader の睡眠時間が結果として返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetAutoOffSleep();
```

2.1.27. GetBaudRateList

関数名	UInt32 GetBaudRateList();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗

関数の説明：

AsReader の現在設定可能なボーレートリストを取得します。

この関数が呼ばれると、その結果は"WhenReceivedBaudRateList" (3.1.19 参照) を介して、現在設定可能な AsReader のボーレートリストが返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetBaudRateList();
```

2.1.28. GetBarcodeTimeout

関数名	UInt32 GetBarcodeTimeout();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗

関数の説明：

AsReader のバーコードタイムアウトを取得します。

この関数が呼ばれると、デリゲート"WhenReceivedBarcodeTimeout" (3.1.15 参照) を通じて、AsReader のバーコードタイムアウトが結果として返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.GetBarcodeTimeout();
```

2.1.29. GetFrequencyList

関数名	UInt32 GetFrequencyList();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-

戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の現在のバンドの利用可能な周波数を取得します。 この関数が呼ばれると、デリゲート "WhenReceivedFrequencyList" (3.1.21 参照) を介して、AsReader の現在のバンドで利用できる周波数を返す結果が返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetFrequencyList();</p>				

2.1.30. GetFrequency

関数名	UInt32 GetFrequency();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の現在の周波数を取得します。 この関数が呼ばれると、その結果はデリゲート "WhenReceivedFrequency" (3.1.22 参照) を通じて返され、AsReader の現在の頻度を返します。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetFrequency();</p>				

2.1.31. RestoreFactorySettings

関数名	UInt32 RestoreFactorySettings();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader を工場出荷時の設定に戻します。 この関数が呼ばれると、デリゲート "WhenReceivedRestoreFactorySettings" (3.1.16 参照)がコールバックされ、AsReader 工場出荷設定に戻す実行結果が返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.RestoreFactorySettings();</p>				

2.1.32. SetBuzzer

関数名	UInt32 SetBuzzer(Types.BuzzerStatus buzzerStatus);			
パラメータ	パラメータ名	イン/アウト	型	説明
	buzzerStatus	In	Types.BuzzerStatus	ブザーの状態 Enum Types.BuzzerStatus (4.1.1 参照)
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のブザー状態を設定します。 この関数が呼ばれると、デリゲート "WhenReceivedSetBuzzerResult "を介して実行結果が返されます (3.1.24 参照)。</p>				

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.SetBuzzer(Types.BuzzerStatus.Off); // ブザーの状態をオフに設定します。
```

2.1.33. SetIdleTime

関数名	UInt32 SetIdleTime(string idleTime);			
パラメータ	パラメータ名	イン/アウト	型	説明
	idleTime	In	string	AsReader のアイドル時間 単位 10ms
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
関数の説明： AsReader のアイドル時間を設定します。 この関数が呼び出されると、デリゲート "WhenReceivedSetIdleTimeResult "を介して実行結果が返される（3.1.26 参照）。				
呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。				
サンプルコード： <pre>Device.SetIdleTime("10"); //AsReader のアイドルタイムを 100ms に設定します。</pre>				

2.1.34. SetAutoOffSleep

関数名	UInt32 SetAutoOffSleep(string autoOffSleepTime);			
パラメータ	パラメータ名	イン/アウト	型	説明

	autoOffSleepTime	In	string	AsReader の自動スリープ時間 単位：秒
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
<p>関数の説明： AsReader のスリープ時間を設定します。 この関数が呼び出されると、デリゲート "WhenReceivedSetAutoOffSleepResult "を介して実行結果が返されます（3.1.28 参照）。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード： Device.SetAutoOffSleep("10"); //AsReader のスリープ時間を 10 秒に設定します。</p>				

2.1.35. SetAutoOffTime

関数名	UInt32 SetAutoOffTime(string autoOffTime);			
パラメータ	パラメータ名	イン/アウト	型	説明
	autoOffTime	In	string	AsReader のオートオフ時間 単位：秒
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
<p>関数の説明： AsReader のオートオフ時間を設定します。 この関数が呼び出されると、デリゲート "WhenReceivedSetAutoOffTimeResult "を介して実行結果が返されます（3.1.29 参照）。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p>				

サンプルコード：

```
Device.SetAutoOffTime("10"); //AsReader のオートオフタイムを 10 秒に設定します。
```

2.1.36. SetQValue

関数名	UInt32 SetQValue(string value);			
パラメータ	パラメータ名	イン/アウト	型	説明
	value	In	string	Q 値 範囲 0~15
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明： AsReader の Q 値を設定します。 この関数が呼ばれると、デリゲート "WhenReceivedSetQValueResult "を介して実行結果が返されます (3.1.25 参照)。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード： Device.SetQValue("4"); // Q 値を 4 に設定する。</p>				

2.1.37. SetBarcodeTimeout

関数名	UInt32 SetBarcodeTimeout(string barcodeTimeout);			
パラメータ	パラメータ名	イン/アウト	型	説明
	barcodeTimeout	In	string	バーコード読み取り時間 単位：秒
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。

				0 : 成功 1: 失敗
<p>関数の説明 : AsReader のバーコード読み取りタイムアウトを設定する。 この関数を呼び出すと、デリゲート "WhenReceivedSetBarcodeTimeoutResult "を介して実行結果が返される (3.1.30 参照)。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.SetBarcodeTimeout("10"); //AsReader のバーコード読み取りタイムアウトを 10 に設定する //seconds(秒)</p>				

2.1.38. SetLinkProfile

関数名	UInt32 SetLinkProfile(string linkProfile);			
パラメータ	パラメータ名	イン/アウト	型	説明
	linkProfile	In	string	リンクプロファイルの値 範囲 0~3
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : AsReader の Link Profile 値を設定します。 この関数が呼び出されると、デリゲート "WhenReceivedSetLinkProfileResult "を介して実行結果が返されます (3.1.27 参照)。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.SetLinkProfile(0); //AsReader のリンクプロファイルを 0 に設定する。</p>				

2.1.39. SetFrequencyAutomatic

関数名	UInt32 SetFrequencyAutomatic();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : AsReader の周波数をホッピングに設定します。 この関数が呼び出されると、デリゲート "WhenReceivedSetFrequencyResult "を介して実行結果が返されます (3.1.23 参照) 。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.SetFrequencyAutomatic();</p>				

2.1.40. SetSession

関数名	UInt32 SetSession(int session);			
パラメータ	パラメータ名	イン/アウト	型	説明
	session	In	int	セッション値 0: S0 1: S1 2: S2 3: S3
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功

				1: 失敗
<p>関数の説明： AsReader の Session 値を設定します。 この関数が呼ばれると、デリゲート "WhenReceivedSetSessionResult "を介して実行結果が返されます (3.1.41 参照)。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード： Device.SetSession(0); //AsReader の Session を S0 に設定します。</p>				

2.1.41. SetInventoryMode

関数名	UInt32 SetInventoryMode(Types.InventoryMode inventoryMode);			
パラメータ	パラメータ名	イン/アウト	型	説明
	inventoryMode	In	Types.InventoryMode	インベントリモード Enum Types.InventoryMode (4.1.2 参照)
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0: 成功 1: 失敗
<p>関数の説明： AsReader のインベントリモードを設定します。 この関数を呼び出すと、デリゲート "WhenReceivedSetInventoryModeResult "を介して実行結果が返される (3.1.44 参照)。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード：</p>				

```
Device.SetInventoryMode(Types.InventoryMode.Single); //AsReader のインベントリモードを
Single に設定します。
```

2.1.42. SetSessionFlag

関数名	UInt32 SetSessionFlag(int sessionFlag);			
パラメータ	パラメータ名	イン/アウト	型	説明
	sessionFlag	In	int	セッションフラグの値 0 : A のみ 1 : B のみ 2: A または B
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : AsReader のセッションフラグを設定します。 この関数が呼ばれると、デリゲート "WhenReceivedSetSessionFlagResult "を介して実行結果が返されます (3.1.43 参照)。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.SetSessionFlag(2); //AsReader のセッションフラグを「A または B」に設定します。</p>				

2.1.43. SetPowerGain

関数名	UInt32 SetPowerGain(int power);			
パラメータ	パラメータ名	イン/アウト	型	説明
	power	In	int	アンテナパワー値

戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のアンテナパワーを設定します。 この関数が呼ばれると、デリゲート "WhenReceivedSetPowerGainResult "を介して実行結果が返されま す (3.1.45 参照) 。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.SetPowerGain(200); // アンテナパワーを 20dB に設定します。</p>				

2.1.44. SetRssiOnOff

関数名	UInt32 SetRssiOnOff(Types.RssiOnOff onOff);			
パラメータ	パラメータ名	イン/アウト	型	説明
	onOff	In	Types.RssiOnOff	RSSI On/Off 設定 Enum Types.RssiOnOff (4.1.3 参照)
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の RSSI On/Off 状態を設定します。 この関数が呼ばれると、デリゲート "WhenReceivedSetRssiOnOffResult "を介して実行結果が返されま す (3.1.48 参照) 。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.SetRssiOnOff(Types.RssiOnOff.Off); // AsReader の RSSI を Off に設定します。</p>				

2.1.45. SetFrequencyAssigned

関数名	UInt32 SetFrequencyAssigned(int[] frequencyIndices);			
パラメータ	パラメータ名	イン/アウト	型	説明
	frequencyIndices	In	Int[]	frequencyList で指定された周波数に対するインデックスのコレクション frequencyList: 3.1.21 を参照。
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明 :

AsReader の周波数を指定した範囲に設定します。

この関数が呼び出されると、デリゲート "WhenReceivedSetFrequencyResult "を介して実行結果が返されます (3.1.23 参照)。

注 : この関数を呼び出すには、まず関数 GetFrequencyList (see: 2.1.29) を呼び出し、現在の周波数帯で利用可能な周波数のリストを取得する必要があります。

呼び出し条件 :

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード :

```
Device.SetFrequencyAssigned(new int[]{ 0, 1}); // AsReader の周波数を、周波数リストの最初の 2 つの周波数の値に設定します。
```

2.1.46. SetBaudRate

関数名	UInt32 SetBaudRate(int baudRateIndex);			
パラメータ	パラメータ名	イン/アウト	型	説明
	baudRateIndex	In	int	baudRateList のインデックス値 (3.1.19 参照)。
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。

				0 : 成功 1 : 失敗
<p>関数の説明 : AsReader のボーレートを設定します。 この関数が呼び出されると、デリゲート "WhenReceivedSetBaudRateResult "を介して実行結果が返されます (3.1.42 参照)。 注意 : この関数を呼び出す前に、関数 GetBaudRateList (2.1.17 参照) を呼び出して AsReader で利用可能なボーレートのリストを取得する必要があります。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.SetBaudRate(0); // baudRateList のインデックスを 0 に設定する。</p>				

2.1.47. SetOperationTime

関数名	UInt32 SetOperationTime(string operationTime);			
パラメータ	パラメータ名	イン/アウト	型	説明
	operationTime	In	string	AsReader の動作時間 単位 : 秒
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の動作時間を設定します。 この関数が呼ばれると、デリゲート "WhenReceivedSetOperationTimeResult "を介して実行結果が返されます (3.1.46 参照)。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード :</p>				

```
Device.SetOperationTime("100"); //AsReader の動作時間を 100 秒に設定します。
```

2.1.48. Kill

関数名	UInt32 Kill(string killPwd);			
パラメータ	パラメータ名	イン/アウト	型	説明
	killPwd	In	string	タグの Kill パスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader でタグをキルします。 この関数が呼ばれると、デリゲート "WhenReceivedKillResult" (3.1.53 参照) がコールバックされ、killing タグアクションの実行結果を返します。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.Kill("444444"); // タグの Kill パスワードは "44444444 "です。</p>				

2.1.49. StartBarcodeScan

関数名	UInt32 StartBarcodeScan();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 :</p>				

AsReader でバーコードの読み取りを開始します。

この関数が呼ばれると、デリゲート "WhenReceivedStartBarcodeScanResult" (3.1.31 参照) がコールバックされ、バーコード読み取り開始アクションの実行結果が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.StartBarcodeScan();
```

2.1.50. StartInventory

関数名	UInt32 StartInventory();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗

関数の説明：

AsReader が RFID タグのインベントリーを開始します。

- 1) この関数が呼ばれると、デリゲート "WhenReceivedStartInventoryResult" (3.1.60 参照) がコールバックされ、インベントリアクション開始時の実行結果が返されます。
- 2) この関数がコールバックされ、AsReader の現在温度が Thredshold を超えると、デリゲート "WhenReceivedWorkTemperature" (3.1.40 参照) がコールバックされて AsReader の現在温度が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.StartInventory();
```

2.1.51. StartInventoryMoisture

関数名	UInt32 StartInventoryMoisture();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader でモイスチャータグのインベントリーを開始します。</p> <p>1) この関数が呼ばれると、デリゲート "WhenReceivedStartInventoryResult" (3.1.60 参照)がコールバックされ、モイスチャータグインベントリー開始アクションの実行結果を返します。</p> <p>2) この関数がコールバックされ、AsReader の現在温度が Thredshold を超えると、デリゲート「WhenReceivedWorkTemperature」(3.1.40 参照) がコールバックされて AsReader の現在温度が返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.StartInventoryMoisture();</p>				

2.1.52. StartInventoryTemperature

関数名	UInt32 StartInventoryTemperature();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 :</p>				

AsReader で温度タグのインベントリを開始します。

1) この関数が呼ばれると、デリゲート "WhenReceivedStartInventoryResult" (3.1.60 参照) がコールバックされ、温度タグインベントリ開始アクションの実行結果が返されます。

2) この関数がコールバックされ、AsReader の現在温度が Thredshold を超えると、デリゲート "WhenReceivedWorkTemperature" (3.1.40 参照) がコールバックされ AsReader の現在温度が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.StartInventoryTemperature();
```

2.1.53. Stop

関数名	UInt32 Stop();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明：

本機能は、AsReader にバーコードの読み取りを停止させたり、タグのインベントリを停止させるために使用します。

この関数が呼ばれると、デリゲート "WhenReceivedStopResult" (3.1.32 参照) がコールバックされ、停止アクションの実行結果が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.Stop();
```

2.1.54. GetEPCMaskData

関数名	UInt32 GetEPCMaskData();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の EPC Mask を取得します。 この関数が呼ばれると、EPC マスクがある場合には、デリゲート "WhenReceivedEPCMaskData" (3.1.37 参照) を介して結果が返され、AsReader の EPC マスクが返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.GetEPCMaskData();</p>				

2.1.55. SetEPCMaskData

関数名	UInt32 SetEPCMaskData(int offset, string mask, int length);			
パラメータ	パラメータ名	イン/アウト	型	説明
	offset	In	int	Mask の開始ビットアドレス
	mask	In	string	マスクデータ
	length	In	int	マスクデータ長
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の EPC Mask を設定します。</p>				

この関数が呼ばれると、結果はデリゲート "WhenReceivedSetEPCMaskResult "を介して返されます (3.1.39 参照)。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.SetEPCMaskData(32, "11112222", 32);
```

2.1.56. ClearEPCMaskData

関数名	UInt32 ClearEPCMaskData();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
<p>関数の説明： AsReader の EPC Mask をクリアします。 この関数が呼ばれると、デリゲート "WhenReceivedClearEPCMaskResult" (3.1.38 参照) がコールバックされ、EPC Macks アクションのクリアの実行結果が返されます。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード： Device.ClearEPCMaskData();</p>				

2.1.57. GetTemperature

関数名	UInt32 GetTemperature();			
パラメータ	パラメータ名	イン/アウト	型	説明

	-	-	-	-
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : AsReader の温度制御パラメータを取得します。 この関数が呼ばれると、デリゲート "WhenReceivedTemperature" (3.1.12 参照) を介して、AsReader の温度制御パラメータが結果として返されることになります。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.Connect(device);</p>				

2.1.58. Read

関数名	UInt32 Read(Types.DataArea dataArea, int offset, int length, string password);			
パラメータ	パラメータ名	イン/アウト	型	説明
	dataArea	In	Types.DataArea	タグの読み取りするデータ領域 Enum Types.DataArea (4.1.5 を参照)
	offset	In	int	タグのデータ領域で読み出すデータの開始アドレス (Words)。
	length	In	int	読み出すデータ長
	password	In	string	タグのアクセスパスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗
<p>関数の説明 : タグアクセスでタグの任意の領域を読み込みます。</p>				

この関数が呼ばれると、デリゲート "WhenReceivedReadResult" (3.1.49 参照) がコールバックされ、タグ読みアクションの実行結果を返します。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.Read(Types.DataArea.EPC, 2, 2, "00000000"); // EPC エリアデータを読み込む。
```

2.1.59. Write

関数名	UInt32 Write(Types.DataArea dataArea, int offset, string writeData, string password);			
パラメータ	パラメータ名	イン/アウト	型	説明
	dataArea	In	Types.DataArea	データを書き込むデータ領域。 Enum Types.DataArea (4.1.5 を参照)
	offset	In	int	タグのデータ領域に書き込むデータの開始アドレス (Words)。
	writeData	In	string	タグに書き込むデータ。
	password	In	string	タグのアクセスパスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0：成功 1：失敗
<p>関数の説明： タグアクセスでタグの任意の領域にデータを書き込みます。 この関数が呼ばれると、デリゲート "WhenReceivedWriteResult" (3.1.50 参照) がコールバックされ、書き込みタグアクションの実行結果が返されます。</p> <p>呼び出し条件： USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード：</p>				

```
Device.Read(Types.DataArea.EPC, 2, "11112222", "00000000"); // EPC 領域にデータを書き込む。
```

2.1.60. Lock

関数名	UInt32 Lock(bool isLockKillPwd, bool isLockAccessPwd, bool isLockEPC, bool isLockTID, bool isLockUser, string password);			
パラメータ	パラメータ名	イン/アウト	型	説明
	isLockKillPwd	In	bool	Kill Password 領域をロックするかどうか。 True: : ロックすること False: 。ロックしない
	isLockAccessPwd	In	bool	アクセスパスワードの領域をロックするかどうか。 True: : ロックすること False: 。ロックしない
	isLockEPC	In	bool	EPC 領域をロックするかどうか。 True: : ロックすること False: 。ロックしない
	isLockTID	In	bool	TID 領域をロックするかどうか。 True: : ロックすること False: 。ロックしない
	isLockUser	In	bool	ユーザーエリアをロックするかどうか。 True: : ロックすること False: 。ロックしない
	password	In	string	タグのアクセスパスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明 :

タグアクセスでタグの任意の領域をロックします。

この関数が呼ばれると、デリゲート "WhenReceivedLockResult" (3.1.51 参照) がコールバックされ、ロックタグ・アクションの実行結果が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.Lock(false, false, true, false, false, "00000000"); // タグの EPC バンクをロックする。
```

2.1.61. PermaLock

関数名	UInt32 PermaLock(bool isLockKillPwd, bool isLockAccessPwd, bool isLockEPC, bool isLockTID, bool isLockUser, string password);			
パラメータ	パラメータ名	イン/アウト	型	説明
	isLockKillPwd	In	bool	タグの Kill Password メモリ領域をロックするかどうか True: : ロックすること Flase: 。ロックしない
	isLockAccessPwd	In	bool	タグのアクセスパスワード記憶領域をロックするかどうか True: : ロックすること Flase: 。ロックしない
	isLockEPC	In	bool	タグの EPC バンクをロックするかどうか True: : ロックすること Flase: 。ロックしない
	isLockTID	In	bool	タグの TID バンクをロックするかどうか True: : ロックすること Flase: 。ロックしない
	isLockUser	In	bool	タグのユーザーバンクをロックするかどうか True: : ロックすること Flase: 。ロックしない
	password	In	string	タグのアクセスパスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗

関数の説明：

タグアクセスでタグの任意の領域を恒久的にロックします。

この関数が呼ばれると、デリゲート "WhenReceivedPermaLockResult" (3.1.52 参照) がコールバックされ、ロックタグアクションの実行結果が返されます。

呼び出し条件：

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード：

```
Device.PermaLock(false, false, true, false, "00000000"); // タグの EPC バンクを恒久的にロックする。
```

2.1.62. UnLock

関数名	UInt32 UnLock(bool isLockKillPwd, bool isLockAccessPwd, bool isLockEPC, bool isLockTID, bool isLockUser,);			
パラメータ	パラメータ名	イン/アウト	型	説明
	isLockKillPwd	In	bool	タグの Kill Password メモリ領域のロックを解除するかどうか True: : ロックを解除すること Flase: 。ロック解除しない
	isLockAccessPwd	In	bool	タグのアクセスパスワード記憶領域をロックするかどうか True: : ロックを解除すること Flase: 。ロック解除しない
	isLockEPC	In	bool	タグの EPC バンクをロックするかどうか True: : ロックを解除すること Flase: 。ロック解除しない
	isLockTID	In	bool	タグの TID バンクをロックするかどうか True: : ロックを解除すること Flase: 。ロック解除しない
	isLockUser	In	bool	タグのユーザーバンクをロックするかどうか

				True: : ロックを解除すること Flase: 。ロック解除しない
	password	In	string	タグのアクセスパスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : タグアクセスでタグの任意の領域をアンロックします。 この関数が呼ばれると、デリゲート "WhenReceivedLockResult" (3.1.51 参照) がコールバックされ、ロック解除タグアクションの実行結果が返されます。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p> <p>サンプルコード : Device.Unlock(false, false, true, false, false, "00000000"); // タグの EPC バンクのロックを解除します。</p>				

2.1.63. SetAccessPassword

関数名	UInt32 SetAccessPassword(string accessPassword);			
パラメータ	パラメータ名	イン/アウト	型	説明
	accessPassword	In	string	タグのアクセスパスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1: 失敗
<p>関数の説明 : タグのアクセスパスワードを設定します。 この関数が呼ばれると、デリゲート "WhenReceivedWriteResult "を介して結果が返されます (3.1.50 参照) 。</p> <p>呼び出し条件 : USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。</p>				

サンプルコード :

```
Device.SetAccessPassword("88888888"); // アクセスパスワードを "88888888" に設定します。
```

2.1.64. SetKillPassword

関数名	UInt32 SetKillPassword(文字列 killPassword);			
パラメータ	パラメータ名	イン/アウト	型	説明
	killPassword	In	string	タグの Kill パスワード
戻り値	-	Out	UInt32	関数を呼び出したときの実行結果。 0 : 成功 1 : 失敗

関数の説明 :

タグの kill パスワードを設定します。

この関数が呼ばれると、デリゲート "WhenReceivedWriteResult "を介して結果が返されます (3.1.50 参照)。

呼び出し条件 :

USB、Bluetooth のいずれかの方式で AsReader と接続している場合、この関数を呼び出すことができます。

サンプルコード :

```
Device.SetKillPassword("44444444"); // タグに "44444444 "の Kill パスワードを設定します。
```

3. AsReaderDelegate クラス

3.1. デリゲート

3.1.1. WhenReceivedFirmwareVersion

デリゲート名	delegate void WhenReceivedFirmwareVersion(string firmwareVersion);			
パラメータ	パラメータ名	イン/アウト	型	説明
	firmwareVersion	In	string	AsReader のファームウェアのバージョン
戻り値	-	-	-	-
<p>デリゲートの説明</p> <p>このデリゲートは、関数 GetFirmwareVersion (2.1.25 参照) が実行されると、コールバックされます。</p> <p>サンプルコード :</p> <pre> WhenReceivedFirmwareVersion whenReceivedFirmwareVersion= ReceivedFirmwareVersion; void ReceivedFirmwareVersion(string firmwareVersion){ // AsReader のファームウェアバージョンを受信した後に実行するプログラム } whenReceivedFirmwareVersion(firmwareVersion); </pre>				

3.1.2. WhenReceivedHardVersion

デリゲート名	delegate void WhenReceivedHardVersion(string hardVersion);			
パラメータ	パラメータ名	イン/アウト	型	説明
	hardVersion	In	string	AsReader のハードウェアバージョン
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 `GetHardVersion` (2.1.9 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedHardVersion whenReceivedHardVersion = ReceivedHardVersion;

void ReceivedHardVersion(string hardVersion){

    // 以下は、AsReader のハードウェアバージョンを受信した後に実行されるプログラム
}

whenReceivedHardVersion(hardVersion);
```

3.1.3. WhenReceivedRegionCountry

デリゲート名	delegate void WhenReceivedRegionCountry(string regionCountry);			
パラメータ	パラメータ名	イン/アウト	型	説明
	regionCountry	In	string	AsReader が適用される地域または国
戻り値	-	-	-	-
デリゲートの説明：				
このデリゲートは、関数 <code>GetRegionCountry</code> (2.1.20 参照) が実行されると、コールバックされます。				
サンプルコード：				
<pre>WhenReceivedRegionCountry whenReceivedRegionCountry = ReceivedRegionCountry; void ReceivedRegionCountry(string regionCountry){ // AsReader が適用される地域や国を受信した後に実行されるプログラム } whenReceivedRegionCountry(regionCountry);</pre>				

3.1.4. WhenReceivedSerialNumber

デリゲート名	delegate void WhenReceivedSerialNumber(string serialNumber);			
パラメータ	パラメータ名	イン/アウト	型	説明

	serialNumber	In	int	AsReader のシリアル番号
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetSerialNumber</code> (2.1.12 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedSerialNumber whenReceivedSerialNumber = ReceivedSerialNumber; void ReceivedSerialNumber(string serialNumber){ // 以下、AsReader のシリアル番号を受け取って実行するプログラム } whenReceivedSerialNumber(serialParam);</pre>				

3.1.5. WhenReceivedRFIDModuleVersion

デリゲート名	delegate void WhenReceivedRFIDModuleVersion(string rfidModuleVersion);			
パラメータ	パラメータ名	イン/アウト	型	説明
	rfidModuleVersion	In	string	AsReader の RFID モジュールバージョン
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetRFIDModuleVersion</code> (2.1.13 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedRFIDModuleVersion whenReceivedRFIDModuleVersion= ReceivedRFIDModuleVersion; void ReceivedRFIDModuleVersion(string rfidModuleVersion) { // RFID モジュール版 AsReader を受信した後に実行するプログラム } whenReceivedRFIDModuleVersion(rfidModuleVersion);</pre>				

3.1.6. WhenReceivedBuzzer

デリゲート名	delegate void WhenReceivedBuzzer(Types.BuzzerStatus buzzerStatus);			
パラメータ	パラメータ名	イン/アウト	型	説明
	buzzerStatus	In	int	AsReader のブザーの状態 Enum です。Types.BuzzerStatus (4.1.1 参照)
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetBuzzer (2.1.16 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedBuzzer whenReceivedBuzzer = ReceivedBuzzer; void ReceivedBuzzer(Types.BuzzerStatus buzzerStatus) { // 以下は、AsReader のブザー状態を受信した後に実行されるプログラム } whenReceivedBuzzer(buzzerStatus);</pre>				

3.1.7. WhenReceivedAutoOffSleep

デリゲート名	delegate void WhenReceivedAutoOffSleep(int autoOffSleep);			
パラメータ	パラメータ名	イン/アウト	型	説明
	autoOffSleep	In	int	AsReader のスリープ時間
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetAutoOffSleep (2.1.26 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedAutoOffSleep whenReceivedAutoOffSleep = ReceivedAutoOffSleep;</pre>				

```
void ReceivedAutoOffSleep(int autoOffSleep){
    // AsReader のスリープ時間を受信した後に実行するプログラム
}
whenReceivedAutoOffSleep(autoOffSleep);
```

3.1.8. WhenReceivedAutoOffTime

デリゲート名	delegate void WhenReceivedAutoOffTime(int autoOffTime);			
パラメータ	パラメータ名	イン/アウト	型	説明
	autoOffTime	In	int	AsReader のオートオフ時間
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetAutoOffTime</code> (2.1.24 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedAutoOffTime whenReceivedAutoOffTime= ReceivedAutoOffTime; void ReceivedAutoOffTime(int autoOffTime){ // AsReader のオートオフ時刻を受信した後に実行するプログラム } whenReceivedAutoOffTime(autoOffTime);</pre></p>				

3.1.9. WhenReceivedQValue

デリゲート名	delegate void WhenReceivedQValue(int QValue);			
パラメータ	パラメータ名	イン/アウト	型	説明
	QValue	In	int	AsReader の Q 値
戻り値	-	-	-	-
<p>デリゲートの説明：</p>				

このデリゲートは、関数 `GetQValue` (2.1.11 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedQValue whenReceivedQValue= ReceivedQValue;

void ReceivedQValue(int QValue){

    // AsReader の Q 値を受信した後に実行するプログラム
}

whenReceivedQValue(QValue);
```

3.1.10. WhenReceivedQMaxValue

デリゲート名	delegate void WhenReceivedQMaxValue(int QMaxValue);			
パラメータ	パラメータ名	イン/アウト	型	説明
	QMaxValue	In	int	AsReader の最大 Q 数
戻り値	-	-	-	-

デリゲートの説明：
このデリゲートは、関数 `GetQMaxValue` (2.1.14 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedQMaxValue whenReceivedQMaxValue= ReceivedQMaxValue;

void ReceivedQMaxValue(int QMaxValue){

    // AsReader の最大 Q を受信した後に実行するプログラム
}

whenReceivedQMaxValue(QMaxValue);
```

3.1.11. WhenReceivedQMinValue

デリゲート名	delegate void WhenReceivedQMinValue(int QMinValue);			
パラメータ	パラメータ名	イン/アウト	型	説明

	QMinValue	In	int	AsReader の最小 Q
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetQMinValue (2.1.10 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedQMinValue whenReceivedQMinValue= ReceivedQMinValue; void ReceivedQMinValue(int QMinValue){ // AsReader の最小 Q を受信した後に実行されるプログラム } whenReceivedQMinValue(QMinValue);</pre>				

3.1.12. WhenReceivedTemperature

デリゲート名	delegate void WhenReceivedTemperature(string currentTemp);			
パラメータ	パラメータ名	イン/アウト	型	説明
	currentTemp	In	String	AsReader の現在の温度
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetTemperature (2.1.57 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedTemperature whenReceivedTemperature = ReceivedTemperature; void ReceivedTemperature(string currentTemp){ // 以下は、AsReader の現在温度を受信した後に実行されるプログラム } ReceivedTemperature(currentTemp);</pre>				

3.1.13. WhenReceivedIdleTime

デリゲート名	delegate void WhenReceivedIdleTime(int idleTime);			
パラメータ	パラメータ名	イン/アウト	型	説明
	idleTime	In	int	AsReader のアイドル時間
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetIdleTime (2.1.4 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedIdleTime whenReceivedIdleTime= ReceivedIdleTime; void ReceivedIdleTime(int idleTime){ // AsReader のアイドル時間を受信した後に実行されるプログラム } whenReceivedIdleTime(idleTime);</pre></p>				

3.1.14. WhenReceivedBattery

デリゲート名	delegate void WhenReceivedBattery(int battery);			
パラメータ	パラメータ名	イン/アウト	型	説明
	battery	In	int	AsReader のバッテリー残量
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetBattery (2.1.17 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedBattery whenReceivedBattery = ReceivedBattery; void ReceivedBattery(int battery){ // AsReader の電池残量を受信した後に実行するプログラム } whenReceivedBattery(battery);</pre></p>				

3.1.15. WhenReceivedBarcodeTimeout

デリゲート名	delegate void WhenReceivedBarcodeTimeout(int barcodeTimeout);			
パラメータ	パラメータ名	イン/アウト	型	説明
	barcodeTimeout	In	int	AsReader のバーコードタイムアウト。
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetBarcodeTimeout (2.1.28 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedBarcodeTimeout whenReceivedBarcodeTimeout = ReceivedBarcodeTimeout; void ReceivedBarcodeTimeout(int barcodeTimeout){ // AsReader のバーコードタイムアウトを受信した後に実行するプログラム } whenReceivedBarcodeTimeout(barcodeTimeout);</pre>				

3.1.16. WhenReceivedRestoreFactorySettings

デリゲート名	delegate void WhenReceivedRestoreFactorySettings();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 RestoreFactorySettings (2.1.31 参照) が実行されると、コールバックされます</p> <p>サンプルコード：</p> <pre>WhenReceivedRestoreFactorySettings whenReceivedRestoreFactorySettings=</pre>				

```

ReceivedRestoreFactorySettings;

void ReceivedRestoreFactorySettings(){

// 以下は、ファクトリーの復元が成功した後に実行されるプログラム

}

whenReceivedRestoreFactorySettings();

```

3.1.17. WhenReceivedLinkProfile

デリゲート名	delegate void WhenReceivedLinkProfile(int linkProfile);			
パラメータ	パラメータ名	イン/アウト	型	説明
	リンクプロファイル	で	イント	AsReader のリンクプロファイル
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 GetLinkProfile (2.1.6 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre> WhenReceivedLinkProfile whenReceivedLinkProfile= ReceivedLinkProfile; void ReceivedLinkProfile(int linkProfile){ // AsReader の Link Profile を受信した後に実行するプログラム } whenReceivedLinkProfile(linkProfile); </pre>				

3.1.18. WhenReceivedBaudRate

デリゲート名	delegate void WhenReceivedBaudRate(int baudRate);			
パラメータ	パラメータ名	イン/アウト	型	説明
	baudRate	In	int	baudRateList のインデックス値

				(3.1.19 参照)。
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetBaudRate</code> (2.1.7 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedBaudRate whenReceivedBaudRate= ReceivedBaudRate; void ReceivedBaudRate(int baudRate){ // 以下は、baudRateList のインデックスを受け取ってから実行されるプログラム } whenReceivedBaudRate(baudRate);</pre></p>				

3.1.19. WhenReceivedBaudRateList

デリゲート名	delegate void WhenReceivedBaudRateList(string[] baudRateList);			
パラメータ	パラメータ名	イン/アウト	型	説明
	baudRateList	In	string[]	AsReader のボーレートリスト
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetBaudRateList</code> (2.1.27 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedBaudRateList whenReceivedBaudRateList = ReceivedBaudRateList; void ReceivedBaudRateList(string[] baudRateList){ // AsReader のボーレートリストを受信して実行するプログラム } whenReceivedBaudRateList(baudRateList);</pre></p>				

3.1.20. WhenReceivedPowerRange

デリゲート名	delegate void WhenReceivedPowerRange(int min, int max);			
パラメータ	パラメータ名	イン/アウト	型	説明
	min	In	int	アンテナパワーの最小値
	max	In	int	アンテナパワーの最大値
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetPowerRange</code> (2.1.15 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedPowerRange whenReceivedPowerRange= ReceivedPowerRange; void ReceivedPowerRange(int min, int max){ // 以下は、アンテナパワーの最大値と最小値を受け取ってから実行するプログラム } whenReceivedPowerRange(min, max);</pre> </p>				

3.1.21. WhenReceivedFrequencyList

デリゲート名	delegate void WhenReceivedFrequencyList(string[] frequencyList);			
パラメータ	パラメータ名	イン/アウト	型	説明
	frequencyList	In	string[]	現在のバンドで使用可能な周波数
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetFrequencyList</code> (2.1.29 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedFrequencyList whenReceivedFrequencyList = ReceivedFrequencyList;</pre> </p>				

```

void ReceivedFrequencyList(string[] frequencyList){
    // 現在のバンドで利用可能な周波数を受信した後に実行されるプログラム
}
whenReceivedFrequencyList(frequencyList);

```

3.1.22. WhenReceivedFrequency

デリゲート名	delegate void WhenReceivedFrequency(int[] frequency);			
パラメータ	パラメータ名	イン/アウト	型	説明
	frequency	In	Int[]	frequencyList で指定された周波数に対するインデックスのコレクション frequencyList: 3.1.21 を参照。
戻り値	-	-	-	-

デリゲートの説明 :

このデリゲートは、関数 GetFrequency (2.1.30 参照) が実行されると、コールバックします。

サンプルコード :

```

WhenReceivedFrequency whenReceivedFrequency= ReceivedFrequency;

void ReceivedFrequency(string frequency){
    // 以下は、AsReader の現在の周波数を受信した後に実行されるプログラ
}
whenReceivedFrequency(frequency);

```

3.1.23. WhenReceivedSetFrequencyResult

デリゲート名	delegate void WhenReceivedSetFrequencyResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 `SetFrequencyAutomatic` (2.1.39 参照) または `SetFrequencyAssigned` (2.1.45 参照) が実行されると、コールバックされます

サンプルコード：

```
WhenReceivedSetFrequencyResult whenReceivedSetFrequencyResult=
ReceivedSetFrequencyResult;

void ReceivedSetFrequencyResult(){
    // 以下は、AsReader の現在の周波数の設定が成功した後に実行されるプログラム
}
whenReceivedSetFrequencyResult();
```

3.1.24. WhenReceivedSetBuzzerResult

デリゲート名	delegate void WhenReceivedSetBuzzerResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 `SetBuzzer` (2.1.32 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedSetBuzzerResult whenReceivedSetBuzzerResult=
ReceivedSetBuzzerResult;

void ReceivedSetBuzzerResult(){
    // 以下は、AsReader のブザーが正常にセットされた後に実行されるプログラム
}
whenReceivedSetBuzzerResult();
```

3.1.25. WhenReceivedSetQValueResult

デリゲート名	delegate void WhenReceivedSetQValueResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 SetQValue (2.1.36 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedSetQValueResult whenReceivedSetQValueResult = ReceivedSetQValueResult; void ReceivedSetQValueResult(){ // AsReader の Q 値の設定に成功した後に実行するプログラム } whenReceivedSetQValueResult();</pre>				

3.1.26. WhenReceivedSetIdleTimeResult

デリゲート名	delegate void WhenReceivedSetIdleTimeResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 SetIdleTime (2.1.33 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedSetIdleTimeResult whenReceivedSetIdleTimeResult=</pre>				

```

ReceivedSetIdleTimeResult;

void ReceivedSetIdleTimeResult(){

    // AsReader のアイドル時間設定に成功した後に実行されるプログラム
}

whenReceivedSetIdleTimeResult();

```

3.1.27. WhenReceivedSetLinkProfileResult

デリゲート名	delegate void WhenReceivedSetLinkProfileResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 SetLinkProfile (2.1.38 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre> WhenReceivedSetLinkProfileResult whenReceivedSetLinkProfileResult= ReceivedSetLinkProfileResult; void ReceivedSetLinkProfileResult(){ // AsReader の Link Profile が正常に設定された後に実行されるプログラム } whenReceivedSetLinkProfileResult(); </pre>				

3.1.28. WhenReceivedSetAutoOffSleepResult

デリゲート名	delegate void WhenReceivedSetAutoOffSleepResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-

戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>SetAutoOffSleep</code> (2.1.34 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedSetAutoOffSleepResult whenReceivedSetAutoOffSleepResult= ReceivedSetAutoOffSleepResult; void ReceivedSetAutoOffSleepResult(){ // AsReader のスリープ時間設定に成功した後に実行されるプログラム } whenReceivedSetAutoOffSleepResult();</pre>				

3.1.29. WhenReceivedSetAutoOffTimeResult

デリゲート名	delegate void WhenReceivedSetAutoOffTimeResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>SetAutoOffTime</code> (2.1.35 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedSetAutoOffTimeResult whenReceivedSetAutoOffTimeResult= ReceivedSetAutoOffTimeResult; void ReceivedSetAutoOffTimeResult(){ // AsReader のオートオフ時間の設定に成功した後に実行されるプログラム } whenReceivedSetAutoOffTimeResult();</pre>				

3.1.30. WhenReceivedSetBarcodeTimeoutResult

デリゲート名	delegate void WhenReceivedSetBarcodeTimeoutResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>SetBarcodeTimeout</code> (2.1.37 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedSetBarcodeTimeoutResult whenReceivedSetBarcodeTimeoutResult = ReceivedSetBarcodeTimeoutResult; void ReceivedSetBarcodeTimeoutResult(){ // AsReader のバーコードタイムアウトが正常に設定された後に実行されるプログラム } whenReceivedSetBarcodeTimeoutResult();</pre></p>				

3.1.31. WhenReceivedStartBarcodeScanResult

デリゲート名	delegate void WhenReceivedStartBarcodeScanResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>StartBarcodeScan</code> (2.1.49 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedStartBarcodeScanResult whenReceivedStartBarcodeScanResult=</pre></p>				

```

ReceivedStartBarcodeScanResult;

void ReceivedStartBarcodeScanResult(){

    // バーコードの読み取りを開始するアクションが正常に実行された後に実行されるプログラム
}

whenReceivedStartBarcodeScanResult();

```

3.1.32. WhenReceivedStopResult

デリゲート名	delegate void WhenReceivedStopResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 Stop (2.1.53 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： WhenReceivedStopResult whenReceivedStopResult= ReceivedStopResult; void ReceivedStopResult(){ // バーコードや RFID タグの読み取りを停止するアクションが正常に実行された後に実行されるプログラム } whenReceivedStopResult();</p>				

3.1.33. WhenReceivedBarcodeData

デリゲート名	delegate void WhenReceivedBarcodeData(byte[] barcodeData, string typeName);			
パラメータ	パラメータ名	イン/アウト	型	説明
	barcodeData	In	byte[]	バーコードデータ

	typeName	In	string	バーコードの種類名
戻り値	-	-	-	-
<p>デリゲートの説明：</p> <p>関数 StartBarcodeScan (2.1.49 参照) を実行して、任意バーコードをスキャンすると、このデリゲートはコールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedBarcodeData whenReceivedBarcodeData= ReceivedBarcodeData; void ReceivedBarcodeData(byte[] barcodeData, string typeName){ // AsReader からバーコードやバーコードタイプの戻り値を受信した後に実行されるプログラム } whenReceivedBarcodeData(barcodeData, typeName);</pre>				

3.1.34. WhenReceivedReadTagData

デリゲート名	delegate void WhenReceivedReadTagData(Types.CommandType commandType, Types.ResultCode resultCode, TagData tagData, string[] readMemory).This void WhenReceivedReadTagData(Types.ResultCode resultCode, TagData tagData, string[] readMemory);			
パラメータ	パラメータ名	イン/アウト	型	説明
	commandType	In	Types.CommandType	コマンドの種類 Enum Types.CommandType (4.1.7 を参照)
	resultCode	In	Types.ResultCode	実行中のコマンドの結果 Enum Types.ResultCode (4.1.6 を参照)
	tagData	In	TagData	読み取ったタグのデータ TagData 型 (5.1 参照)。
	readMemory	In	string[]	AsReader で読み込んだタグメモリーデータ
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 Read (2.1.58 参照) または Write (2.1.59 参照) または Lock (2.1.60 参照) または PermaLock (2.1.61 参照) または UnLock (2.1.62 参照) を実行すると、コールバックされます。

サンプルコード：

```
WhenReceivedReadTagData whenReceivedReadTagData= ReceivedReadTagData;

void ReceivedReadTagData(Types.CommandType commandType, Types.ResultCode
resultCode, TagData tagData, string[] readMemory){

    // 以下、AsReader でタグの読み書き、ロック、キルを行い、返されたデータを処理するためのプログラム
}

whenReceivedReadTagData(commandType, resultCode, tagData, readMemory);
```

3.1.35. WhenReceivedData

デリゲート名	delegate void WhenReceivedData(byte[] data);			
パラメータ	パラメータ名	イン/アウト	型	説明
	data	In	byte[]	AsReader からのデータ返却
戻り値	-	-	-	-
デリゲートの説明：				
AsReader から何らかのデータが返されると、このデリゲートがコールバックされます。				
サンプルコード：				
WhenReceivedData whenReceivedData= ReceivedData;				
void ReceivedData(byte[] data){				
// AsReader から何らかのデータが返ってきた後に実行されるプログラム				
}				
whenReceivedData(data);				

3.1.36. WhenReceivedTagData

デリゲート名	delegate void WhenReceivedTagData(TagData tag);			
パラメータ	パラメータ名	イン/アウト	型	説明
	tag	In	TagData	タグデータ TagData 型 (5.1 参照)。
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 StartInventory (2.1.50 参照) または StartInventoryMoisture (2.1.51 参照) または StartInventoryTemperature (2.1.52 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedTagData whenReceivedTagData= ReceivedTagData; void ReceivedTagData(TagData tag){ // AsReader から RFID データの返送を受けた後に実行するプログラム } whenReceivedTagData(tag);</pre> </p>				

3.1.37. WhenReceivedEPCMaskData

デリゲート名	delegate void WhenReceivedEPCMaskData(int maskOffset, string maskData, int maskLength);			
パラメータ	パラメータ名	イン/アウト	型	説明
	maskOffset	In	int	マスクオフセットアドレス
	maskData	In	string	マスクデータ
	maskLength	In	int	マスクデータの長さ
戻り値	-	-	-	-
<p>デリゲートの説明：</p>				

このデリゲートは、関数 `GetEPCMaskData` (2.1.54 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedEPCMaskData whenReceivedEPCMaskData= ReceivedEPCMaskData;

void ReceivedEPCMaskData(int maskOffset, string maskData, int maskLength){

    // AsReader の EPC Mask を受信した後に実行するプログラム
}

whenReceivedEPCMaskData(maskOffset, maskData, maskLength);
```

3.1.38. WhenReceivedClearEPCMaskResult

デリゲート名	delegate void WhenReceivedClearEPCMaskResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>ClearEPCMaskData</code> (2.1.56 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedClearEPCMaskResult whenReceivedClearEPCMaskResult= ReceivedClearEPCMaskResult; void ReceivedClearEPCMaskResult(){ // EPC Mask を全てクリアするアクションが正常に実行された後に実行されるプログラム } whenReceivedClearEPCMaskResult();</pre> </p>				

3.1.39. WhenReceivedSetEPCMaskResult

デリゲート名	delegate void WhenReceivedSetEPCMaskResult();			
パラメータ	パラメータ名	イン/アウト	型	説明

		ウト		
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>ClearEPCMaskData</code> (2.1.56 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedSetEPCMaskResult whenReceivedSetEPCMaskResult= ReceivedSetEPCMaskResult; void ReceivedSetEPCMaskResult(){ // AsReader の EPC Mask が正常に設定された後に実行されるプログラム } whenReceivedSetEPCMaskResult();</pre>				

3.1.40. WhenReceivedWorkTemperature

デリゲート名	デリゲート <code>void WhenReceivedWorkTemperature(string workTemp, string threshold);</code>			
パラメータ	パラメータ名	イン/アウト	型	説明
	workTemp	In	String	AsReader の現在の温度
	threshold	In	String	温度閾値
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>StartInventory</code> (2.1.50 参照) または <code>StartInventoryMoisture</code> (2.1.51 参照) または <code>StartInventoryTemperature</code> (2.1.52 参照) が実行されて AsReader の温度が閾値を超えると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedWorkTemperature whenReceivedWorkTemperature = ReceivedWorkTemperature;</pre>				

```
void ReceivedWorkTemperature (string workTemp, string threshold){
    // 以下、AsReader の現在温度と閾値を受け取ってから実行するプログラム
}
ReceivedWorkTemperature(workTemp, threshold);
```

3.1.41. WhenReceivedSetSessionResult

デリゲート名	delegate void WhenReceivedSetSessionResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>SetSession</code> (2.1.40 参照) が実行されると、コールバックされます。</p> <p>サンプルコード：</p> <pre>WhenReceivedSetSessionResult whenReceivedSetSessionResult= ReceivedSetSessionResult; void ReceivedSetSessionResult(){ // AsReader の Session が正常に設定された後に実行されるプログラム } whenReceivedSetSessionResult();</pre>				

3.1.42. WhenReceivedSetBaudRateResult

デリゲート名	delegate void WhenReceivedSetBaudRateResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 `SetBaudRate` (2.1.46 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedSetBaudRateResult whenReceivedSetBaudRateResult =
ReceivedSetBaudRateResult;

void ReceivedSetBaudRateResult(){
    // AsReader のボーレートが正常に設定された後に実行されるプログラム
}
whenReceivedSetBaudRateResult();
```

3.1.43. WhenReceivedSetSessionFlagResult

デリゲート名	delegate void WhenReceivedSetSessionFlagResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 `SetSessionFlag` (2.1.42 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedSetSessionFlagResult whenReceivedSetSessionFlagResult=
ReceivedSetSessionFlagResult;

void ReceivedSetSessionFlagResult(){
    // AsReader の Session Flag が正常にセットされた後に実行されるプログラム
}
whenReceivedSetSessionFlagResult();
```

3.1.44. WhenReceivedSetInventoryModeResult

デリゲート名	delegate void WhenReceivedSetInventoryModeResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 SetInventoryMode (2.1.41 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： WhenReceivedSetInventoryModeResult whenReceivedSetInventoryModeResult= ReceivedSetInventoryModeResult; void ReceivedSetInventoryModeResult(){ // AsReader のインベントリモードが正常に設定された後に実行されるプログラム }</p>				

3.1.45. WhenReceivedSetPowerGainResult

デリゲート名	delegate void WhenReceivedSetPowerGainResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 SetPowerGain (2.1.43 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： WhenReceivedSetPowerGainResult whenReceivedSetPowerGainResult= ReceivedSetPowerGainResult;</p>				

```
void ReceivedSetPowerGainResult(){
    // AsReader のアンテナパワーが正常に設定された後に実行されるプログラム
}
whenReceivedSetPowerGainResult();
```

3.1.46. WhenReceivedSetOperationTimeResult

デリゲート名	delegate void WhenReceivedSetOperationTimeResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 SetOperationTime (2.1.47 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： WhenReceivedSetOperationTimeResult whenReceivedSetOperationTimeResult= ReceivedSetOperationTimeResult; void ReceivedSetOperationTimeResult(){ // AsReader の動作時間設定に成功した後に実行されるプログラム } whenReceivedSetOperationTimeResult();</p>				

3.1.47. WhenReceivedSetInventoryTimeResult

デリゲート名	delegate void WhenReceivedSetInventoryTimeResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 `SetInventoryTime` (2.1.48 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedSetInventoryTimeResult whenReceivedSetInventoryTimeResult=
ReceivedSetInventoryTimeResult;

void ReceivedSetInventoryTimeResult(){
    // AsReader のインベントリタイムの設定が成功した後に実行されるプログラム
}
whenReceivedSetInventoryTimeResult();
```

3.1.48. WhenReceivedSetRssiOnOffResult

デリゲート名	delegate void WhenReceivedSetRssiOnOffResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 `SetRssiOnOff` (2.1.44 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedSetRssiOnOffResult whenReceivedSetRssiOnOffResult=
ReceivedSetRssiOnOffResult;

void ReceivedSetRssiOnOffResult(){
    // AsReader の RSSI On/Off が正常に設定された後に実行されるプログラム
}
whenReceivedSetRssiOnOffResult();
```

3.1.49. WhenReceivedReadResult

デリゲート名	delegate void WhenReceivedReadResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 Read (2.1.58 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： WhenReceivedReadResult whenReceivedReadResult= ReceivedReadResult; void ReceivedReadResult(){ // タグの読み取りが成功した後に実行されるプログラム }</p>				

3.1.50. WhenReceivedWriteResult

デリゲート名	delegate void WhenReceivedWriteResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 Write (2.1.59 参照) または SetAccessPassword (2.1.63 参照) または SetKillPassword (2.1.64 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： WhenReceivedWriteResult whenReceivedWriteResult= ReceivedWriteResult; void ReceivedWriteResult(){</p>				

```
// 書き込みタグのアクションが正常に実行された後に実行されるプログラム
}
whenReceivedWriteResult();
```

3.1.51. WhenReceivedLockResult

デリゲート名	delegate void WhenReceivedLockResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：
このデリゲートは、関数 Lock (2.1.60 参照) または UnLock (2.1.62 参照) が実行されると、コールバックされます。

サンプルコード：

```
WhenReceivedLockResult whenReceivedLockResult= ReceivedLockResult;
void ReceivedLockResult(){
    // ロック・アンロックタグのアクションが正常に実行された後に実行されるプログラム
}
whenReceivedLockResult();
```

3.1.52. WhenReceivedPermaLockResult

デリゲート名	delegate void WhenReceivedPermaLockResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-

デリゲートの説明：
このデリゲートは、関数 PermaLock (2.1.61 参照) が実行されると、コールバックされます。

```

サンプルコード：
WhenReceivedPermaLockResult                                whenReceivedPermaLockResult=
ReceivedPermaLockResult;

void ReceivedPermaLockResult(){

    // 永久ロックタグのアクションが正常に実行された後に実行されるプログラム
}
whenReceivedPermaLockResult();

```

3.1.53. WhenReceivedKillResult

デリゲート名	delegate void WhenReceivedKillResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
デリゲートの説明： このデリゲートは、関数 Kill (2.1.48 参照) が実行されると、コールバックされます。				
サンプルコード： WhenReceivedKillResult whenReceivedKillResult = ReceivedKillResult; void ReceivedKillResult(){ // ここで、は、killing タグアクションが正常に実行された後に実行されるプログラム } whenReceivedKillResult();				

3.1.54. WhenReceivedOperationTime

デリゲート名	delegate void WhenReceivedOperationTime(int operationTime);			
パラメータ	パラメータ名	イン/アウト	型	説明
	operationTime	In	int	AsReader の動作時間

戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetOperationTime</code> (2.1.21 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedOperationTime whenReceivedOperationTime= ReceivedOperationTime; void ReceivedOperationTime(){ // AsReader の動作時間を受信した後に実行するプログラム } whenReceivedOperationTime();</pre></p>				

3.1.55. WhenReceivedInventoryMode

デリゲート名	delegate void WhenReceivedInventoryMode(Types.InventoryMode mode);			
パラメータ	パラメータ名	イン/アウト	型	説明
	mode	In	Types.InventoryMode	AsReader のインベントリモード Enum Types.InventoryMode (4.1.2 参照)
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 <code>GetInventoryMode</code> (2.1.22 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： <pre>WhenReceivedInventoryMode whenReceivedInventoryMode= ReceivedInventoryMode; void ReceivedInventoryMode(Types.InventoryMode mode){ // AsReader のインベントリモードを受信した後に実行するプログラム } whenReceivedInventoryMode(mode);</pre></p>				

3.1.56. WhenReceivedRssiOnOff

デリゲート名	delegate void WhenReceivedRssiOnOff(Types.RssiOnOff OnOff);			
パラメータ	パラメータ名	イン/アウト	型	説明
	OnOff	In	Types.RssiOnOff	AsReader の RSSI オン/オフ状態 Enum Types.RssiOnOff (4.1.3 参照)
戻り値	-	-	-	-
<p>デリゲートの説明 : このデリゲートは、関数 GetRssiOnOff (2.1.8 参照) が実行されると、コールバックされます。</p> <p>サンプルコード : <pre>WhenReceivedRssiOnOff whenReceivedRssiOnOff= ReceivedRssiOnOff; void ReceivedRssiOnOff(Types.RssiOnOff OnOff){ // AsReader の RSSI On/Off 状態を受信した後に実行するプログラム } whenReceivedRssiOnOff(OnOff);</pre></p>				

3.1.57. WhenReceivedPowerGain

デリゲート名	delegate void WhenReceivedPowerGain(int power);			
パラメータ	パラメータ名	イン/アウト	型	説明
	power	In	int	AsReader のアンテナパワー
戻り値	-	-	-	-
<p>デリゲートの説明 : このデリゲートは、関数 GetPower (2.1.23 参照) が実行されると、コールバックされます。</p> <p>サンプルコード : <pre>WhenReceivedPowerGain whenReceivedPowerGain = ReceivedPowerGain; void ReceivedPowerGain(int power){ // 以下は、AsReader のアンテナパワーを受信した後に実行するプログラム } </pre></p>				

```
whenReceivedPowerGain(power);
```

3.1.58. WhenReceivedSession

デリゲート名	delegate void WhenReceivedSession(int session);			
パラメータ	パラメータ名	イン/アウト	型	説明
	session	In	int	AsReader のセッション
戻り値	-	-	-	-
<p>デリゲートの説明 : このデリゲートは、関数 <code>GetSession</code> (2.1.18 参照) が実行されると、コールバックされます。</p> <p>サンプルコード : <pre>WhenReceivedSession whenReceivedSession= ReceivedSession; void ReceivedSession(int session){ // AsReader の Session を受信した後に実行されるプログラム } whenReceivedSession(session);</pre> </p>				

3.1.59. WhenReceivedSessionFlag

デリゲート名	delegate void WhenReceivedSessionFlag(int sessionFlag);			
パラメータ	パラメータ名	イン/アウト	型	説明
	session Flag	In	int	AsReader のセッションフラグ
戻り値	-	-	-	-
<p>デリゲートの説明 : このデリゲートは、関数 <code>GetSessionFlag</code> (2.1.19 参照) が実行されると、コールバックされます。</p> <p>サンプルコード : <pre>WhenReceivedSessionFlag whenReceivedSessionFlag= ReceivedSessionFlag;</pre> </p>				

```
void ReceivedSessionFlag(int sessionFlag)。
    // AsReader の Session Flag を受信した後に実行されるプログラム
}
whenReceivedSessionFlag(sessionFlag);
```

3.1.60. WhenReceivedStartInventoryResult

デリゲート名	delegate void WhenReceivedStartInventoryResult();			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>デリゲートの説明： このデリゲートは、関数 StartInventory (2.1.50 参照) または StartInventoryMoisture (2.1.51 参照) または StartInventoryTemperature (2.1.52 参照) が実行されると、コールバックされます。</p> <p>サンプルコード： WhenReceivedStartInventoryResult whenReceivedStartInventoryResult= ReceivedStartInventoryResult; void ReceivedStartInventoryResult(){ // RFID タグのインベントリを開始するアクションが正常に実行された後に実行されるプログラム } whenReceivedStartInventoryResult();</p>				

3.1.61. WhenReceivedTriggerStatus

デリゲート名	delegate void WhenReceivedTriggerStatus(Types.ButtonActionStatus buttonActionStatus);			
パラメータ	パラメータ名	イン/アウト	型	説明
	buttonActionStatus	In	Types.ButtonActionStatus	AsReaderのトリガーボタンの状態

				Enum Types.ButtonActionStatus (4.1.9を参照)
戻り値	-	-	-	-
<p>デリゲートの説明： AsReader の Trigger ボタンが押されると、このデリゲートがコールバックされます。</p> <p>サンプルコード： WhenReceivedTriggerStatus whenReceivedTriggerStatus = ReceivedTriggerStatus; void ReceivedTriggerStatus(Types.ButtonActionStatus buttonActionStatus){ // AsReader のトリガーボタンが押されたり離されたりしている状態を受信した後に実行されるプログラム } whenReceivedTriggerStatus(buttonActionStatus);</p>				

3.1.62. WhenReceivedModeStatus

デリゲート名	delegate void WhenReceivedModeStatus(Types.ButtonActionStatus buttonActionStatus);			
パラメータ	パラメータ名	イン/アウト	型	説明
	buttonActionStatus	In	Types.ButtonActionStatus	AsReader の Mode ボタンの状態 Enum Types.ButtonActionStatus (4.1.9を参照)
戻り値	-	-	-	-
<p>デリゲートの説明： AsReader の Mode ボタンが押されると、このデリゲートがコールバックされます。</p> <p>サンプルコード： WhenReceivedModeStatus whenReceivedModeStatus= ReceivedModeStatus; void ReceivedModeStatus(Types.ButtonActionStatus buttonActionStatus){</p>				

```

// AsReader のモードボタンが押されたり離されたりしている状態を受信した後に実行されるプログラム
}
whenReceivedModeStatus(buttonActionStatus);

```

3.1.63. WhenDeviceIsConnected

デリゲート名	delegate void WhenDeviceIsConnected(bool isConnected);			
パラメータ	パラメータ名	イン/アウト	型	説明
	isConnected	In	bool	接続状態 true: 接続中 false: 未接続
戻り値	-	-	-	-

デリゲートの説明：

Bluetooth 接続の場合、このデリゲートは関数 "Connect" (2.1.5 参照) または "Disconnect" (2.1.2 参照) を実行して AsReader との接続状態を変更すると、コールバックされます。

サンプルコード：

```

WhenDeviceIsConnected whenDeviceIsConnected = DeviceIsConnected;
void DeviceIsConnected(bool isConnected){
    // AsReader との接続状態が変更された後に実行されるプログラム
}
whenDeviceIsConnected(isConnected);

```

4. タイプクラス

4.1. 列挙型

4.1.1. BuzzerStatus

ブザーの状態

定義	説明
Off	ブザーオフ
Low	低音量
High	高音量

4.1.2. InventoryMode

インベントリーモード

定義	説明
Single	シングルタグ・インベントリーモード
Continuous	連続インベントリーモード

4.1.3. RssiOnOff

RSSI オン/オフ状態

定義	説明
Off	RSSI データは出力されません。
On	RSSI データが出力されます。

4.1.4. Baudrate

ボーレート

定義	説明
_9600	9600bps
_19200	9600bps
_115200	115200 bps
_230400	230400 bps
_460800	460800bps

4.1.5. DataArea

データ領域

定義	説明
Reserved	リザーブドバンク
EPC	EPC バンク
TID	TID バンク
User	ユーザーバンク

4.1.6. ResultCode

AsReader が命令実行後に返す値

定義	説明
NoError	成功
OtherError	その他のエラー
Undefined	未定義エラー
MemoryOverrun	メモリーオーバーラン
MemoryLocked	メモリーロック
InsufficientPower	電力不足

HandleMismatch	ハンドル不整合
CRCErrror	CRC エラー
TagLost	タグロスト
CommandFormatError	コマンドフォーマットエラー
OutOfRetries	リトライ回数不足

4.1.7. CommandType

コマンドタイプ

定義	説明
ReadMemory	タグメモリ読み出しコマンド
WriteMemory	タグメモリ書き込みコマンド

4.1.8. TagType

タグタイプ

定義	説明
Moisture	モイスチャータグ
Temperature	温度タグ

4.1.9. ButtonActionStatus

ボタンの動作状態

定義	説明
Up	ボタンを解放
Down	ボタンを押下

4.1.10. BluetoothType

Bluetooth タイプ

定義	説明
BluetoothSPP	SPP

5. 構造物

5.1. TagData

タグデータパラメータ

変数名	型	商品説明
EPC	string	タグの EPC 値
PC	string	タグの PC 値
TagType	Types.TagType	タグタイプ Enum Types.TagType (4.1.8 を参照)
Frequency	string	タグの周波数 単位 KHz
Moisture	int	タグのモイスター値
Temperature	double	タグの温度値
Phase	double	タグの位相値
Rssi	double	タグの RSSI 値

6. AsBluetoothManager クラス

6.1. 関数

6.1.1. StartSearching

関数名	void StartSearching(Types.BluetoothType type);			
パラメータ	パラメータ名	イン/アウト	型	説明
	type	In	Types.BluetoothType	Bluetooth タイプ Enum BluetoothType (4.1.10 参照)
戻り値	-	-	-	-
<p>関数の説明 : AsReaders の検索を開始します。 この関数が呼ばれると、取得した AsReaders のリストが AsBluetoothManager クラスの "WhenReceivedDeviceList" (1 参照) デリゲートを介してコールバックされます。</p> <p>呼び出し条件 : Bluetooth で AsReader とペアリングされている状態であること。</p> <p>サンプルコード : asBluetoothManager.StartSearching(Types.BluetoothType.BluetoothSPP);</p>				

6.1.2. StopSearching

関数名	void StopSearching()。			
パラメータ	パラメータ名	イン/アウト	型	説明
	-	-	-	-
戻り値	-	-	-	-
<p>関数の説明 :</p>				

AsReaders の検索を停止します。

呼び出し条件：

1. Bluetooth で AsReader とペアリングされている状態であること。
2. 関数 StartSearching (6.1.1 参照) が呼び出されデバイス検索中の状態であること。

サンプルコード：

```
asBluetoothManager.StopSearching();
```

6.2. デリゲート

6.2.1. WhenReceivedDeviceList

デリゲート名	delegate void WhenReceivedDeviceList(List<BluetoothDevice> devices);			
パラメータ	パラメータ名	イン/アウト	型	説明
	devices	In	List<BluetoothDevice>	検索された AsReader の一覧 BluetoothDevice タイプ(7.1 参照)
戻り値	-	-	-	-

デリゲートの説明：

このデリゲートは、関数 StartSearching (6.1.1 参照) が実行され、任意の AsReader が検索されると、コールバックされます。

サンプルコード：

```
WhenReceivedDeviceList whenReceivedDeviceList = ReceivedDeviceList;

Void ReceivedDeviceList(List<BluetoothDevice> devices){

    // 以下、検索した AsReader のリストを受け取ってから実行するプログラム
}

whenReceivedDeviceList(devices);
```

7. BluetoothDevice クラス

7.1. プロパティ

物件名	物件紹介	型	説明
DeviceBluetoothType	get/set	Types.BluetoothType	AsReader の Bluetooth タイプ Enum BluetoothType (4.1.10 参照)
DeviceName	get/set	String	AsReader の Bluetooth 接続名
DeviceID	get/set	String	AsReader の MAC アドレス