M3 MOBILE

WR15 SDK 메뉴얼

V1.2

2025/11/21

Revision History

Date	Version	Comments	Author	Reviewer
2025/10/21	V1.0	Initial	Seongrak_Choi	
2025/11/13	V1.1	Changed SDK distribution	Seongrak_Choi	
2025/11/21	V1.2	Add Error Code	Seongrak_Choi	

Index

1. Import SDK

1.1 Libraries

1. **setting.gradle.kts** 에 maven 경로를 추가해 주세요.

```
dependencyResolutionManagement {
    repositories {
        google()
        mavenCentral()

        //이 줄 추가
        maven("https://m3mobile.github.io/wr15-sdk-maven")
    }
}
```

* **Groovy** 를 사용하시는 경우

```
dependencyResolutionManagement {
    repositories {
        google()
        mavenCentral()

        //이 줄 추가

        maven { url "https://m3mobile.github.io/wr15-sdk-maven" }
    }
}
```

2. **build.gradle(Module:app)**에 알맞은 Version 을 import 해주세요.

```
dependencies {
    implementation("com.m3:wr15-sdk:1.2.0") //원하시는 버전을 추가해 주세요.
    ...
}
```

2. Permissions Required

저희 SDK 를 사용하기 위해서는 필수적으로 블루투스와 위치 권한이 필요합니다. 자사에 알맞게 타겟팅 하고 있는 안드로이드 버전에 따라 적절한 권한을 등록해 주세요.

2.1 Android 12+(API 31+)이 중심인 경우

```
<uses-permission android:name="android.permission.BLUETOOTH_SCAN"
    android:usesPermissionFlags="neverForLocation"/>
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>
```

2.2 Android 6~11(API25~30)도 지원해야 하는 경우

```
<!-- Android 6~11 (API 25~30) -->
<uses-permission android:name="android.permission.BLUETOOTH"
   android:maxSdkVersion="30"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
   android:maxSdkVersion="30"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
   android:maxSdkVersion="30"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
   android:maxSdkVersion="30"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
   android:maxSdkVersion="30"/>

<!-- Android 12+ (API 31+) -->
<uses-permission android:name="android.permission.BLUETOOTH_SCAN"
   android:usesPermissionFlags="neverForLocation"/>
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>
```

2.3 런타임 권한 요청 확인 필수

- BLUETOOTH_SCAN

- BLUETOOTH_CONNECT

- ACCESS_FINE_LOCATION

- ACCESS_COARSE_LOCATION

위 4 개의 권한의 경우, 앱 사용자에게 직접 허용을 받아야 하는 런타임 권한이기 때문에, SDK 사용 전 블루투스 권한 허용 여부에 대해 주의해 주세요.

3. Initialize SDK

3.1 Initialize the dependence libraries

라이브러리를 사용하시기 전에, initialize 로 꼭 초기화를 진행해 주세요.

Kotlin

```kotlin
val config = M3Wr15SdkConfig.builder()
            .connectionType(ConnectionType.SPP)
            .enableLog(true)
            .build()


M3Wr15Sdk.initialize(context, config)

```

Java

```java
M3Wr15SdkConfig config = M3Wr15SdkConfig.builder()
        .connectionType(ConnectionType.SPP)
        .enableLog(true)
        .build();

M3Wr15Sdk sdk =
        M3Wr15Sdk.initialize(context, config);
```

3.1.1 M3Wr15SdkConfig

- *connectionType( )* : WR15 와의 통신 방식

*** ConnectionType.BLE 는 v1.0 에서 지원되지 않습니다.**

- *enableLog( )* : SDK 의 로그 출력 여부

3.1.2 Context

Context는 스캔 및 SharedPreference에 사용되기 때문에, **Activity Context가 아닌 Application Context를 전달**하세요.

4. Search for WR15

4.1 S2P (Scan to Pair)

WR15 장비로 S2P 바코드를 스캔하여, 스캔한 장비를 검색합니다.

4.1.1 Generate Barcode

```
/* M3Wr15Sdk */

@JvmOverloads
@JvmStatic
fun getBarcodeBitmapForS2P(width: Int = 260, height: Int = 50): Bitmap?

@JvmStatic
fun getCmdForS2P(): String
```

일반적인 상황에서는 *getBarcodeBitmapForS2P( )* 메소드로부터 반환 받은 비트맵을 이용하여 S2P 바코드를 구현하시면 되지만, 다른 바코드 타입을 원하시는 경우에는 *getCmdForS2P( )*의 반환 String 값을 이용하여 원하시는 바코드타입으로 만들어서 사용하시면 됩니다.

4.1.2 Start S2P Scan

WR15 링 스캐너가 위에서 생성한 바코드를 스캔하는 순간에 S2P 를 위한 S2P Scan 이 진행되고 있어야 합니다.

For Kotlin

```
/* M3Wr15Sdk */

@RequiresPermission(
allOf = [Manifest.permission.BLUETOOTH_SCAN, Manifest.permission.BLUETOOTH_CONNECT])
fun startS2PScan(
    onDeviceFound: (BluetoothDevice) -> Unit,
    onScanFailed: (errorCode: Int) -> Unit
)
```

For Java

```
/* M3Wr15Sdk */

@JvmStatic
@RequiresPermission(
allOf = [Manifest.permission.BLUETOOTH_SCAN, Manifest.permission.BLUETOOTH_CONNECT])
fun startS2PScan(listener: S2PScanResultListener)

```

```
interface S2PScanResultListener {
    fun onDeviceFound(device: BluetoothDevice)
    fun onScanFailed(errorCode: Int)
}
```

startS2PScan( ) 메소드는 **BLUETOOTH_SCAN, BLUETOOTH_CONNECT** 권한 허용이 필수이기 때문에 호출 전, 권한 허용 여부를 확인하는 방어코드를 사용하시길 추천드립니다.

4.1.3 Stop S2P Scan

WR15 링 스캐너를 찾은 직후는 *startS2PScan( )*을 중단하셔야 합니다. S2P 바코드가 존재하는 화면을 벗어나는 경우에는 s*tartS2PScan( )*을 중단하는 것이 바람직합니다.

```
/* M3Wr15Sdk */

@RequiresPermission(Manifest.permission.BLUETOOTH_SCAN)
fun stopS2PScan()
```

Example

```
@RequiresPermission(Manifest.permission.BLUETOOTH_SCAN)
fun stopS2PScan()

M3Wr15Sdk.startS2PScan(object: S2PScanResultListener) {
    override fun onDeviceFound(device: BluetoothDevice) {
        M3Wr15Sdk.stopS2PScan() //스캔 중단
        // 발견한 BluetoothDevice 로 연결 시도
    }

    override fun onScanFailed(errorCode: Int) {
        // 스캔 실패 처리
    }
})
```

4.2 Search and Connect

주변의 WR15 스캐너 디바이스들을 검색합니다.

4.2.1 Start Discovery Scan

Discovery Scan 으로 동작하기 때문에 스캔 시작 후, 약 20 초 뒤에 자동으로 스캔 종료됩니다.

```
/*M3Wr15Sdk*/

@RequiresPermission(allOf = [Manifest.permission.BLUETOOTH_SCAN,
Manifest.permission.BLUETOOTH_CONNECT])
fun startDiscoveryScan(listener: DiscoveryScanResultListener)

```

```
interface DiscoveryScanResultListener {
    fun onStarted() //스캔 시작
    fun onError() // 에러 발생
    fun onDeviceFound(device: DiscoveryDeviceModel) //WR15 스캐너 장치 발견
    fun onFinished() //스캔 종료
}
```

Exmaple

```
M3Wr15Sdk.startDiscoveryScan(object : DiscoveryScanResultListener {
    @RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
    override fun onDeviceFound(device: DiscoveryDeviceDodel) {
    // 디바이스 발견 처리
    }

    override fun onError(errorMsg: String) {
    // 에러 처리
    }

    override fun onFinished() {
    // 스캔 완료 처리
    }

    override fun onStarted() {
    // 스캔 시작 처리
    }
}
)
```

4.2.2 Stop Discovery Scan

Discovery Scan 을 중지합니다.

```
/*M3Wr15Sdk*/

@JvmStatic
fun stopDiscoveryScan()
```

5. Connect to WR15

5.1 Connect

BluetoothDevice 장비를 SDK 와 연결합니다.

For Kotlin

```kotlin
/*M3Wr15Sdk*/

@RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
fun connectToDevice(
    device: BluetoothDevice,
    onSuccess: (session: TransportSession) -> Unit,
    onFailure: (errorCode: Int, errorMsg: String) -> Unit
)
```

For Java

```kotlin
/*M3Wr15Sdk*/

@JvmStatic
@RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
fun connectToDevice(device: BluetoothDevice, listener: ConnectResultListener)

interface ConnectResultListener {
    fun onSuccess(session: TransportSession)
    fun onFailure(errorCode:Int, errorMsg: String)
}
```

onSuccess(session: TransportSession)로 반환받는 TransportSession 을 이용하여 추후 WR15 링 스캐너와 통신할 수 있습니다.

Example

```kotlin
val sdk = M3Wr15Sdk.initialize(applicationContext, config)

sdk.startS2PScan(object: DeviceFindResultListener {
    override fun onDeviceFound(device: BluetoothDevice) {
        sdk.stopS2PScan()
        sdk.connectToDevice(device, object:ConnectResultListener {
            override fun onSuccess(session: TransportSession) {
                //성공 처리
            }
            override fun onFailure(errorMsg: String) {
                //실패 처리
            }
        })
    }

    override fun onScanFailed(errorCode: Int) {
        // 스캔 실패 처리
```

```
    }
})
```

6. Disconnect

6.1 Disconnect to WR15

연결되어 있는 WR15 스캐너와의 연결을 해제합니다.

```
/* TransportSession */

//For Kotlin
fun disconnect(onSuccess: () -> Unit, onFailure: (errorMsg: String) -> Unit)

//For Java
fun disconnect(listener: DisconnectResultListener)
```

6.2 Observe Disconnect

연결이 해제되는 순간을 감지합니다.

```
/* TransportSession */

fun addDisconnectListener(disconnectListener: DisconnectListener)
fun removeDisconnectListener(disconnectListener: DisconnectListener)

fun interface DisconnectListener {
    fun onDisconnect()
}
```

Example

```
val disconnectListener = object: DisconnectListener {
    override fun onDisconnect() {
    // 연결 해제 시 처리
    }
  }
fun connectToDevice() {
    sdk.connectToDevice(device, object:ConnectResultListener {
        override fun onSuccess(session:TransportSession) {
            session.addDisconnectListener(disconnectListener)
        }
        override fun onFailure(errorMsg: String) {
            //실패 처리
        }
    })
}
```

```
16.
17. fun disconnect() {
18.     if (sdk.isTransportSessionInitialized()) { // 세션이 초기화된 경우에만 연결 해제 시도
19.         val session = sdk.getTransportSession()
20.         session .disconnect(
21.             onSuccess = {
22.                 // 연결 해제 성공 처리
23.                 session.removeDisconnectListener(disconnectListener)
24.             },
25.             onFailure = { errorMsg ->
26.                 // 연결 해제 실패 처리
27.             }
28.         )
29.     }
30. }
```

메모리 누수를 방지하기 위하여 연결이 끊길 때나 Listener 가 더 이상 사용되지 않을 때, *removeDisconnectListener( )*를 이용하여, 꼭 등록하신 리스너는 해제해 주세요.

* addDisconnectListener 에 등록했던 같은 객체의 리스너를 removeDisconnectListener 의 파라미터로 넣어야 합니다.

7. Receive Barcode Data

7.1 Add Barcode Data

WR15 링 스캐너로부터 디코딩 된 바코드 데이터를 수신하기 위해서는 Listener 를 등록해야 합니다.

```
/* TransportSession */

fun addDecodingDataListener(decodingDataListener: DecodingDataListener)

interface DecodingDataListener {
    fun onReceive(decodingData: ByteArray)
}
```

7.2 Remove Barcode Data

BarcodeDataListener 를 해제합니다.

```
/* TransportSession */

fun removeBarcodeDataListener(barCodeDataListener: BarcodeDataListener)
```

메모리 누수를 방지하기 위해, 연결이 끊길 때나, 더 이상 Listener 를 사용하지 않을 때는 꼭 해제해 주세요.

* addBarcodeDataListener 에 등록했던 같은 객체의 리스너를 removeBarcodeDataListener 의 파라미터로 넣어야 합니다.

8. Scanner Device Information

WR15 링 스캐너에 대한 디바이스 및 스캐너 설정 정보는 ***data class DeviceState( )*** 형태로 제공됩니다.
DeviceState 의 구체적인 정보는 9.DeviceState 을 참조해 주세요.

8.1 Observe DeviceState

DeviceState 는 Callback 형식으로 제공되며, 설정이 변경될 때 마다, 재전달 됩니다.

```
/* TransportSession */

fun addDeviceStateListener(listener: DeviceStateListener)
fun removeDeviceStateListener(listener: DeviceStateListener)

interface DeviceStateListener {
    fun onDeviceStateChanged(state: DeviceState)
}

data class DeviceState(
    val deviceInfo: DeviceInfo = DeviceInfo(),
    val general: GeneralSettings = GeneralSettings(),
    val symbologies: Symbologies = Symbologies(),
    val basicFormat: BasicFormat = BasicFormat(),
    val readerSettings: ReaderSettings = ReaderSettings(),
    val devSettings: DevSettings = DevSettings()
)
```

8.2 Set Scanner Setting

WR15 스캐너의 설정을 변경합니다.

For Kotlin

```

fun setSetting(
    setting: InternalSettingCommand,
    onSuccess: () -> Unit,
    onFailure: (errorCode: Int, errorMsg: String) -> Unit,
)
```

For Java

```
/* TransportSession */

fun setSetting(setting: SettingCommand, listener: SetSettingResultListener)

```

```
interface SetSettingResultListener{
    fun onSuccess()
    fun onFailure(errorCode: Int, errorMsg: String)
}
```

8.2.1 SettingCommand

SettingCommand 는 *setSetting( )*의 파라미터로 이용되며, 변경하고자 하는 설정과 옵션을 담는 객체입니다. *SettingCommand* 생성은 *Settings* 객체를 이용합니다.

8.2.2 Generate SettingCommand

Settings 는 SettingsCommand 를 손쉽게 생성하도록 돕는 API 진입점입니다.

```
object Settings {
    object Codabar {
        @JvmStatic
        fun setEnable(enable: Boolean): SettingCommand
        @JvmStatic
        fun setLength1(length: Int): SettingCommand
        ...
    }

    object Code11 {
        @JvmStatic
        fun setEnable(enable: Boolean): SettingCommand
        @JvmStatic
        fun setLength1(length: Int): SettingCommand
        ...
     }

     object General {
        @JvmStatic
        fun findMe(): SettingCommand
        @JvmStatic
        fun setAimer(enable: Boolean): SettingCommand
        @JvmStatic
        fun setIllumination(enable: Boolean): SettingCommand
        @JvmStatic
        fun setSound(type: SoundType): SettingCommand
        @JvmStatic
        fun setVibrate(enable: Boolean): SettingCommand
        ...
       }
     ... //다른 설정들
}
```

더 자세한 설정 항목은 10.Settings 를 참조해 주세요.

Example

For Kotlin

```kotlin
session.setSetting(
    Settings.EAN13.setEnable(true),
    onSuccess = {
      // 설정 성공 처리
    },
    onFailure = { errorMsg ->
      // 설정 실패 처리
    }
)
```

For Java

```java
session.setSetting(
    Settings.General.setSound(SoundType.OFF),
    object : SetSettingResultListener {
        override fun onFailure(errorMsg: String) {
          // 설정 실패 처리
        }

        override fun onSuccess() {
          // 설정 성공 처리
        }
    }
)
```

8.2.3 Caution

WR15 링 스캐너 종류(ex. E4770, SE4107)마다 설정할 수 있는 설정과 옵션값이 조금 다릅니다. 문서를 잘 확인 후, 스캐너 종류에 알맞은 설정만 요청할 수 있도록 주의해 주세요.

사용중인 WR15 링 스캐너에 지원하지 않는 설정을 요청한 경우, *setSetting* 의 Callback 이 *onFailure( )*로 처리됩니다.

9. DeviceState

DeviceState는 스캐너의 모든 정보를 담는 객체입니다. 사용하시는 WR15 스캐너가 지원하지 않는 옵션의 경우 값은 null 또는 Empty String입니다.

Type형은 10. Settings 에서 확인하실 수 있습니다.

9.1 Strucutre

```
data class DeviceState(
    val deviceInfo: DeviceInfo = DeviceInfo(),
    val general: GeneralSettings = GeneralSettings(),
    val symbologies: Symbologies = Symbologies(),
    val basicFormat: BasicFormat = BasicFormat(),
    val readerSettings: ReaderSettings = ReaderSettings(),
    val devSettings: DevSettings = DevSettings()
)
```

9.1.1 DeviceInfo

```
data class DeviceInfo(
    val name: String? = null,
    val macAddress: String? = null,
    val model: String? = null,
    val vendor: VendorType? = null,
    val serialNumber: String? = null,
    val fwVersion: String? = null,
    val releaseDate: String? = null,
    val batteryPercent: Int? = null,
    val scannerId: ScannerIdType? = null,
    val btMode: BtModeType? = null,
)
```

9.1.2 GeneralSettings

```
data class GeneralSettings(
    val sound: SoundType? = null,
    val vibrateEnabled: Boolean? = null,
    val ledEnabled: Boolean? = null,
    val buttonEnabled: Boolean? = null,
    val aimEnabled: Boolean? = null,
    val illuminationEnabled: Boolean? = null,
)
```

9.1.3 Symbologies

```
data class Symbologies(
    val australianPostalVals: AustralianVals = AustralianVals(),
    val aztecVals: AztecVals = AztecVals(),
    val chinese2Of5Vals: Chinese2Of5Vals = Chinese2Of5Vals(),
```

```
    val codabarVals: CodabarVals = CodabarVals(),
    val code11Vals: Code11Vals = Code11Vals(),
    val code39Vals: Code39Vals = Code39Vals(),
    val code93Vals: Code93Vals = Code93Vals(),
    val code128Vals: Code128Vals = Code128Vals(),
    val compositeVals: CompositeVals = CompositeVals(),
    val dataMatrixVals: DataMatrixVals = DataMatrixVals(),
    val discrete2Of5Vals: Discrete2Of5Vals = Discrete2Of5Vals(),
    val dotCodeVals: DotCodeVals = DotCodeVals(),
    val ean8Vals: Ean8Vals = Ean8Vals(),
    val ean13Vals: Ean13Vals = Ean13Vals(),
    val gs1DataBar14Vals: Gs1DataBar14Vals = Gs1DataBar14Vals(),
    val gs1DataBarExpandedVals: Gs1DataBarExpandedVals = Gs1DataBarExpandedVals(),
    val gs1DataBarLimitedVals: Gs1DataBarLimitedVals = Gs1DataBarLimitedVals(),
    val gs1128Vals: Gs1128Vals = Gs1128Vals(),
    val hanXinVals: HanXinVals = HanXinVals(),
    val interleaved2Of5Vals: Interleaved2Of5Vals = Interleaved2Of5Vals(),
    val isbt128Vals: Isbt128Vals = Isbt128Vals(),
    val japanesePostalVals: JapanesePostalVals = JapanesePostalVals(),
    val korean3Of5Vals: Korean3Of5Vals = Korean3Of5Vals(),
    val matrix2Of5Vals: Matrix2Of5Vals = Matrix2Of5Vals(),
    val maxiCodeVals: MaxiCodeVals = MaxiCodeVals(),
    val microPdf417Vals: MicroPdf417Vals = MicroPdf417Vals(),
    val microQrCodeVals: MicroQrCodeVals = MicroQrCodeVals(),
    val msiVals: MsiVals = MsiVals(),
    val netherlandsKixVals: NetherlandsKixVals = NetherlandsKixVals(),
    val pdf417Vals: Pdf417Vals = Pdf417Vals(),
    val qrCodeVals: QrCodeVals = QrCodeVals(),
    val ukPostalVals: UkPostalVals = UkPostalVals(),
    val upcAVals: UpcAVals = UpcAVals(),
    val upcE1Vals: UpcE1Vals = UpcE1Vals(),
    val upcEanVals: UpcEanVals = UpcEanVals(),
    val upcEVals: UpcEVals = UpcEVals(),
    val upuFicsPostalVals: UpuFicsPostalVals = UpuFicsPostalVals(),
    val usPlanetVals: UsPlanetVals = UsPlanetVals(),
    val usPostnetVals: UsPostnetVals = UsPostnetVals(),
)
```

AustralianVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class AustralianVals(
    val enabled: Boolean? = null,
)
```

AztecVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class AztecVals(
    val enabled: Boolean? = null
)
```

Chinese2Of5Vals

지원하는 Scanner : ***SE4107, SE5500***

```
data class Chinese2Of5Vals(
    val enabled: Boolean? = null
)
```

CodabarVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class CodabarVals(
    val enabled: Boolean? = null,
    val length1: Int? = null,
    val length2: Int? = null,
    val clsiEditing: Boolean? = null, //E4770 미지원
    val notisEditing: Boolean? = null,
    val securityLevel: CodabarType.SecurityType? = null, //E4770 미지원
)
```

Code11Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Code11Vals(
    val enabled: Boolean? = null,
    val length1: Int? = null,
    val length2: Int? = null,
    val reportCheckDigit: Boolean? = null,
    val verifyCheckDigit: Code11Type.VerifyCheckDigitType? = null
)
```

Code39Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Code39Vals(
    val enabled: Boolean? = null,
    val triopticCode39: Boolean? = null, //E4770 미지원
    val length1: Int? = null,
    val length2: Int? = null,
    val reducedQuietZone: Boolean? = null, //E4770 미지원
    val convertToCode32: Boolean? = null,
    val fullAscii: Boolean? = null,
    val reportCheckDigit: Boolean? = null, // true = TRANSMIT
    val reportCode32Prefix: Boolean? = null,
    val securityLevel: Code39Type.SecurityType? = null, //E4770 미지원
    val verifyCheckDigit: Boolean? = null  // true = ENABLE
)
```

Code93Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Code93Vals(
    val enabled: Boolean? = null,
    val length1: Int? = null,
```

```
    val length2: Int? = null
)
```

Code128Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Code128Vals(
    val enabled: Boolean? = null,
    val length1: Int? = null,
    val length2: Int? = null,
    val reducedQuietZone: Boolean? = null, //E4770 미지원
    val ignoreFnc4: Boolean? = null, //E4770 미지원
    val checkIsbtTable: Boolean? = null, //E4770 미지원
    val isbt128ConcatMode: Code128Type.ISBT128ConcatModeType? = null, //E4770 미지원
    val securityLevel: Code128Type.SecurityType? = null, //E4770 미지원
    val emulationMode: Boolean? = null//E4770 미지원
)
```

CompositeVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class CompositeVals(
    val enabled: Boolean? = null, //SE4107, SE5500 미지원
    val compositeABEnabled: Boolean? = null, //E4770 미지원
    val compositeCEnabled: Boolean? = null, //E4770 미지원
    val tlc39Enable: Boolean? = null, //E4770 미지원
    val upcCompositeMode: CompositeType.UPCCompositeModeType? = null //E4770 미지원
)
```

E4770 에서는 Enabled 하나의 설정에 ABEnabled, CEnabled 2 개가 통합되어 있습니다. SE4107, SE5500 에서 ABEnabled 와 CEnabled 를 true 로 설정하면 enabled = true 와 같은 동작을 합니다.

DataMatrixVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class DataMatrixVals(
    val enabled: Boolean? = null,
    val inverse: DataMatrixType.InverseType? = null,
    val decodeMirrorImages: DataMatrixType.DecodeMirrorImagesType? = null, //E4770 미지원
    val gs1DataMatrix: Boolean? = null //E4770 미지원
)
```

Discrete2Of5Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Discrete2Of5Vals(
    val enabled: Boolean? = null,
```

```
    val length1: Int? = null,
    val length2: Int? = null
)
```

DotCodeVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class DotCodeVals(
    val enabled: Boolean? = null,
    val inverse: DotCodeType.InverseType? = null,
    val mirror: DotCodeType.MirrorType? = null,
    val prioritize: Boolean? = null
)
```

Ean8Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Ean8Vals(
    val enabled: Boolean? = null
)
```

Ean13Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Ean13Vals(
    val enabled: Boolean? = null
)
```

Gs1DataBar14Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Gs1DataBar14Vals(
    val enabled: Boolean? = null
)
```

Gs1DataBarExpandedVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class Gs1DataBarExpandedVals(
    val enabled: Boolean? = null
)
```

Gs1DataBarLimitedVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class Gs1DataBarLimitedVals(
    val enabled: Boolean? = null,
```

```
    val securityLevel: Gs1DataBarLimitedType.SecurityType? = null
)
```

Gs1128Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Gs1128Vals(
    val enabled: Boolean? = null
)
```

HanXinVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class HanXinVals(
    val enabled: Boolean? = null,
    val inverse: HanXinType.InverseType? = null
)
```

Interleaved2Of5Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Interleaved2Of5Vals(
    val enabled: Boolean? = null,
    val length1: Int? = null,
    val length2: Int? = null,
    val checkDigit: Interleaved2Of5Type.CheckDigitType? = null,
    val reportCheckDigit: Boolean? = null,     // true = TRANSMIT
    val securityLevel: Interleaved2Of5Type.SecurityType? = null, //E4770 미지원
    val convertItf14ToEan13: Boolean? = null, //E4770 미지원
    val reducedQuietZone: Boolean? = null //E4770 미지원
)
```

Isbt128Vals

지원하는 Scanner : ***SE4107, SE5500***

```
data class Isbt128Vals(
    val enabled: Boolean? = null
)
```

JapanesePostalVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class JapanesePostalVals(
    val enabled: Boolean? = null
)
```

Korean3Of5Vals

지원하는 Scanner : ***SE4107, SE5500***

```
data class Korean3Of5Vals(
    val enabled: Boolean? = null
)
```

Matrix2Of5Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class Matrix2Of5Vals(
    val enabled: Boolean? = null,
    val length1: Int? = null,
    val length2: Int? = null,
    val redundancy: Boolean? = null, //E4770 미지원
    val reportCheckDigit: Boolean? = null,  // true = TRANSMIT
    val verifyCheckDigit: Boolean? = null   // true = ENABLE
)
```

MaxiCode

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class MaxiCodeVals(
    val enabled: Boolean? = null
)
```

MicroPdf417Vals

지원하는 Scanner : ***SE4107, SE5500***

```
data class MicroPdf417Vals(
    val enabled: Boolean? = null
)
```

MicroQrCodeVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class MicroQrCodeVals(
    val enabled: Boolean? = null
)
```

MsiVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class MsiVals(
    val enabled: Boolean? = null,
    val length1: Int? = null,
    val length2: Int? = null,
    val checkDigit: MsiType.CheckDigitType? = null,
    val checkDigitScheme: MsiType.CheckDigitSchemeType? = null,
    val reportCheckDigit: Boolean? = null                    // true = TRANSMIT
```

```
)
```

NetherlandKixVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class NetherlandsKixVals(
    val enabled: Boolean? = null
)
```

Pdf417Vals

지원하는 Scanner : ***SE4107, SE5500***

```
data class Pdf417Vals(
    val enabled: Boolean? = null
)
```

QrCodeVals

지원하는 Scanner **: *SE4107, SE5500, E4770***

```
data class QrCodeVals(
    val enabled: Boolean? = null
)
```

UkPostalVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class UkPostalVals(
    val enabled: Boolean? = null,
    val reportCheckDigit: Boolean? = null // true = TRANSMIT
)
```

UpcAVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class UpcAVals(
    val enabled: Boolean? = null,
    val reportCheckDigit: Boolean? = null,     // true = TRANSMIT
    val preamble: UpcAType.PreambleType? = null,
    val upcA2BitsSupplementals: Boolean? = null,
    val upcA5BitsSupplementals: Boolean? = null
)
```

UpcE1Vals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class UpcE1Vals(
    val enabled: Boolean? = null,
    val convertToUpcA: Boolean? = null,// true = CONVERT //E4770 미지원
    val preamble: UpcE1Type.PreambleType? = null, //E4770 미지원
    val reportCheckDigit: Boolean? = null // true = TRANSMIT //E4770 미지원
)
```

UPpcEanVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class UpcEanVals(
    val reportEan8CheckDigit: Boolean? = null, // true = TRANSMIT
    val reportEan13CheckDigit: Boolean? = null, // true = TRANSMIT

    val supplementalMode: UpcEanType.SupplementalModeType? = null, //E4770 미지원
    val supplementalAimIdFormat: UpcEanType.SupplementalAimIdFormatType? = null, //E4770 미지원
    val reducedQuietZone: Boolean? = null, //E4770 미지원
    val bookland: UpcEanType.BooklandType? = null, //E4770 미지원
    val uccCouponExtend: Boolean? = null, //E4770 미지원
    val couponReport: UpcEanType.CouponReportType? = null, //E4770 미지원
    val issnEan: Boolean? = null, //E4770 미지원
    val translateUpcAToEan13: Boolean? = null, //E4770 미지원
    val supplementalRedundancy: Int? = null, //E4770 미지원
    val supplemental2: Boolean? = null, //E4770 미지원
    val supplemental5: Boolean? = null, //E4770 미지원

    val ean8ReadSupplements: Boolean? = null, //SE4107, SE5500 미지원
    val ean13ReadSupplements: Boolean? = null, // //SE4107, SE5500 미지원
    val ean8Supplemental2: Boolean? = null, // //SE4107, SE5500 미지원
    val ean13Supplemental2: Boolean? = null, // //SE4107, SE5500 미지원
    val ean8Supplemental5: Boolean? = null, // //SE4107, SE5500 미지원
    val ean13Supplemental5: Boolean? = null // //SE4107, SE5500 미지원
)
```

SE4107, **SE5500** 에서 지원하는 *Supplemental2* 의 기능은 E4770 에서 *ean8Supplemental2*, *ean8Supplemental* 5 로 나뉘어져 있고, *Supplemental5* 의 기능은 *ean13Supplemental2*, *ean13Supplemental5* 로 나뉘어져 있습니다.

추가적으로, E4770 에서 *ean8Supplemental2*, *ean8Supplemental* 5 설정을 사용하시려면 *ean8ReadSupplements* 설정을 true 로 설정해 주셔야 합니다. 마찬가지로 *ean13Supplemental2*, *ean13Supplemental5* 설정을 사용하시려면, ean13ReadSupplements 설정을 true 로 설정해 주셔야 합니다.

UpcEVals

지원하는 Scanner : ***SE4107, SE5500, E4770***

```
data class UpcEVals(
    val enabled: Boolean? = null,
    val convertToUpcA: Boolean? = null,   // true = CONVERT
    val preamble: UpcEType.PreambleType? = null,
    val reportCheckDigit: Boolean? = null, // true = TRANSMIT
    val upcE2BitsSupplementals: Boolean? = null,
    val upcE5BitsSupplementals: Boolean? = null
```

```
)
```

UpuFicsPostalVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class UpuFicsPostalVals(
    val enabled: Boolean? = null
)
```

UsPlanetVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class UsPlanetVals(
    val enabled: Boolean? = null,
    val reportCheckDigit: Boolean? = null // true = TRANSMIT
)
```

UsPostnetVals

지원하는 Scanner : ***SE4107, SE5500***

```
data class UsPostnetVals(
    val enabled: Boolean? = null
)
```

9.1.4 BasicFormat

```
data class BasicFormat(
    val transmitCodeId: BasicFormatType.TransmitCodeIdType? = null,
    val endChar: BasicFormatType.EndCharType? = null,
    val substringFormation: String? = null,
    val removeFnc: Boolean? = null,
    val translateData: String? = null,
    val prefixPostfixAsAsciiHex: Boolean? = null,
    val prefix: String? = null,
    val postfix: String? = null
)
```

9.1.5 readerSettings

```
data class ReaderSettings(
    val readMode: ReaderType.ReadModeType? = null,
    val laserOnTime: Int? = null,
    val inverse1D: ReaderType.Inverse1DType? = null,
    val quietZoneLevelFor1D: ReaderType.QuietZoneLevelType? = null,
    val poorQualityDecodeEffort: ReaderType.PoorQualityDecodeEffortType? = null,
)
```

9.1.6 DevSettings

```
data class DevSettings(
    val batteryCallback: Boolean? = null
)
```

10. Settings

*Settings*는 각종 스캐너 설정에 대한 *SettingCommand*를 손쉽게 생성할 수 있는 SDK의 진입점입니다. 바코드 심볼로지별 옵션과 공통 리더기 옵션은 일관된 메서드 형태로 랩핑하여, 호출만으로 안전한 커맨드를 만들 수 있습니다.

10.1 AustralianPostal

```
object AustralianPostal {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.2 Aztec

```
object Aztec {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.3 Chinese2of5

```
object Chinese2of5 {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.4 Codabar

```
object Codabar {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand

    @JvmStatic //E4770 미지원
    fun setCLSIEditing(enable: Boolean): SettingCommand

    @JvmStatic
    fun setNotisEditing(enable: Boolean): SettingCommand
```

```
16.
17.     @JvmStatic //E4770 미지원
18.     fun setSecurityLevel(value: CodabarType.SecurityType): SettingCommand
19. }
```

10.4.1 CodabarType

```
1.  object CodabarType {
2.      enum class SecurityType : TypeValue {
3.          LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
4.      }
5.  }
```

10.5 Code11

```
1.  object Code11 {
2.      @JvmStatic
3.      fun setEnable(enable: Boolean): SettingCommand
4.
5.      @JvmStatic
6.      fun setLength1(length: Int): SettingCommand
7.
8.      @JvmStatic
9.      fun setLength2(length: Int): SettingCommand
10.
11.     @JvmStatic
12.     fun setReportCheckDigit(enable: Boolean): SettingCommand
13.
14.     @JvmStatic
15.     fun setVerifyCheckDigit(value: Code11Type.VerifyCheckDigitType): SettingCommand
16. }
```

10.5.1 Code11Type

```
1.  object Code11Type {
2.      enum class VerifyCheckDigitType : TypeValue {
3.          DISABLE,
4.          ONE_CHECK_DIGIT,
5.          TWO_CHECK_DIGITS
6.      }
7.  }
```

10.6 Code39

```
1.  object Code39 {
2.      @JvmStatic
3.      fun setEnable(enable: Boolean): SettingCommand
4.
5.      @JvmStatic //E4770 미지원
6.      fun setTriopticCode39(enable: Boolean): SettingCommand
7.
```

```
    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand

    @JvmStatic //E4770 미지원
    fun setReducedQuietZone(enable: Boolean): SettingCommand

    @JvmStatic
    fun setConvertToCode32(enable: Boolean): SettingCommand

    @JvmStatic
    fun setFullAscii(enable: Boolean): SettingCommand

    @JvmStatic
    fun setReportCheckDigit(enable: Boolean): SettingCommand

    @JvmStatic
    fun setReportCode32Prefix(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setSecurityLevel(value: Code39Type.SecurityType): SettingCommand

    @JvmStatic
    fun setVerifyCheckDigit(enable: Boolean): SettingCommand
}
```

10.6.1 Code39Type

```
object Code39Type {
    enum class SecurityType : TypeValue {
        LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
    }
}
```

10.7 Code93

```
object Code93 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand
}
```

10.8 Code128

```
object Code128 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand

    @JvmStatic //E4770 미지원
    fun setReducedQuietZone(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setIgnoreFnc4(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setCheckIsbtTable(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setIsbt128ConcatMode(value: Code128Type.ISBT128ConcatModeType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setSecurityLevel(value: Code128Type.SecurityType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setEmulationMode(enable: Boolean): SettingCommand
}
```

10.8.1 Code128Type

```
object Code128Type {
    enum class SecurityType : TypeValue {
        LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
    }

    enum class ISBT128ConcatModeType : TypeValue {
        ENABLE, DISABLE, AUTO
    }
}
```

10.9 Composite

```
object Composite {
    @JvmStatic //SE4107, SE5500 미지원
    fun setEnable(enable: Boolean): SettingCommand //SE4107, SE5500 미지원

    @JvmStatic //E4770 미지원
    fun setABEnable(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setCEnable(value: Boolean): SettingCommand

```

```
    @JvmStatic //E4770 미지원
    fun setTlc39Enable(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setUpcCompositeMode(value: CompositeType.UPCCompositeModeType): SettingCommand
}
```

- *setEnable(enable: Boolean)* 설정은 *setABEnable(enable)*, *setCEnable(enable)* 두 설정을 동시에 적용합니다. 즉, E4770 스캐너의 경우, *setEnable(true)*는 AB 와 C 를 모두 활성화, *setEnbale(false)*는 둘 다 비활성화합니다.

10.9.1 CompositeType

```
object CompositeType {
    enum class UPCCompositeModeType : TypeValue {
        NEVER_LINKED, ALWAYS_LINKED, AUTO_DISCRIMINATE
    }
}
```

10.10 DataMatrix

```
object DataMatrix {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setInverse(value: DataMatrixType.InverseType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setDecodeMirrorImages(value: DataMatrixType.DecodeMirrorImagesType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setGs1DataMatrix(enable: Boolean): SettingCommand
}
```

10.10.1 DatatMatrixType

```
object DataMatrixType {
    enum class InverseType : TypeValue {
        REGULAR_ONLY, INVERSE_ONLY, AUTO
    }

    enum class DecodeMirrorImagesType : TypeValue {
        NEVER, ALWAYS, AUTO
    }
}
```

10.11 Discrete2Of5

```
object Discrete2Of5 {
```

```
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand
}
```

10.12 DotCode

```
object DotCode {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setInverse(value: DotCodeType.InverseType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setMirror(value: DotCodeType.MirrorType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setPrioritize(enable: Boolean): SettingCommand
}
```

10.12.1 DotCodeType

```
object DotCodeType {
    enum class InverseType : TypeValue { ENABLE, DISABLE, AUTODETECT }
    enum class MirrorType : TypeValue { REGULAR, INVERSE_ONLY, AUTODETECT }
}
```

10.13 EAN8

```
object EAN8 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.14 EAN13

```
object EAN13 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.15 Gs1DataBar14

```
object Gs1DataBar14 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.16 Gs1DataBarExpanded

```
object Gs1DataBarExpanded {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.17 Gs1DataBarLimited

```
object Gs1DataBarLimited {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setLimitedSecurityLevel(value: Gs1DataBarLimitedType.SecurityType): SettingCommand
}
```

10.17.1 Gs1DataBarLimitedType

```
object Gs1DataBarLimitedType {
    enum class SecurityType : TypeValue {
        LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
    }
}
```

10.18 Gs1128

```
object Gs1128 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.19 HanXin

```
object HanXin {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setInverse(value: HanXinType.InverseType): SettingCommand
}
```

10.20 Interleaved2Of5

```
object Interleaved2Of5 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand

    @JvmStatic
    fun setCheckDigit(value: Interleaved2Of5Type.CheckDigitType): SettingCommand

    @JvmStatic
    fun setReportCheckDigit(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setSecurityLevel(value: Interleaved2Of5Type.SecurityType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setConvertItf14ToEan13(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setReducedQuietZone(enable: Boolean): SettingCommand
}
```

10.20.1 Interleaved2Of5Type

```
object Interleaved2Of5Type {
    enum class CheckDigitType : TypeValue {
        DISABLE,
        ENABLE,
        USS_CHECK_DIGIT, //E4770 미지원
        OPCC_CHECK_DIGIT //E4770 미지원
    }

    enum class SecurityType : TypeValue {
        LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
    }
}
```

10.21 ISBT128

```
object ISBT128 {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.22 JapanesePostal

```
object JapanesePostal {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.23 Korean3of5

```
object Korean3of5 {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.24 Matrix2of5

```
object Matrix2Of5 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand

    @JvmStatic //E4770 미지원
    fun setRedundancy(enable: Boolean): SettingCommand

    @JvmStatic
    fun setReportCheckDigit(enable: Boolean): SettingCommand

    @JvmStatic
    fun setVerifyCheckDigit(enable: Boolean): SettingCommand
}
```

10.25 MaxiCode

```
object MaxiCode {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.26 MicroPdf417

```
object MicroPdf417 {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.27 MicroQrCode

```
object MicroQrCode {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.28 MSI

```
object MSI {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLength1(length: Int): SettingCommand

    @JvmStatic
    fun setLength2(length: Int): SettingCommand

    @JvmStatic
    fun setCheckDigit(value: MsiType.CheckDigitType): SettingCommand

    @JvmStatic
    fun setCheckDigitScheme(value: MsiType.CheckDigitSchemeType): SettingCommand

    @JvmStatic
    fun setReportCheckDigit(enable: Boolean): SettingCommand
}
```

10.28.1 MsiType

```
object MsiType {
    enum class CheckDigitType : TypeValue {
        ONE_CHECK_DIGIT, TWO_CHECK_DIGITS
    }

    enum class CheckDigitSchemeType : TypeValue {
        MOD_10_10, MOD_10_11
    }
}
```

10.29 NetherlandsKix

```
object NetherlandsKix {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.30 Pdf417

```
object Pdf417 {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.31 QrCode

```
object QrCode {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand
}
```

10.32 UKPostal

```
object UKPostal {
    @JvmStatic //E4770 미지원
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setReportCheckDigit(enable: Boolean): SettingCommand
}
```

10.33 UPCA

```
object UPCA {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setReportCheckDigit(enable: Boolean): SettingCommand

    @JvmStatic
    fun setPreamble(value: UpcAType.PreambleType): SettingCommand

```

```
    @JvmStatic
    fun setSupplemental2Bit(enable: Boolean): SettingCommand

    @JvmStatic
    fun setSupplemental5Bit(enable: Boolean): SettingCommand
}
```

10.33.1 UpcAType

```
object UpcAType {
    enum class PreambleType : TypeValue {
        NONE, SYS_CHAR, COUNTRY_SYS_CHAR
    }
}
```

10.34 UPCE1

```
object UPCE1 {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setConvertToUpcA(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setPreamble(value: UpcE1Type.PreambleType): SettingCommand

    @JvmStatic //E4770 미지원
    fun setReportCheckDigit(enable: Boolean): SettingCommand
}
```

10.34.1 UpcE1Type

```
object UpcE1Type {
    enum class PreambleType : TypeValue {
        NONE, SYS_CHAR, COUNTRY_SYS_CHAR
    }
}
```

10.35 UPCEan

```
object UPCEan {
    @JvmStatic
    fun setReportEan8CheckDigit(enable: Boolean): SettingCommand

    @JvmStatic
    fun setReportEan13CheckDigit(enable: Boolean): SettingCommand

    @JvmStatic //E4770 미지원
    fun setSupplementalMode(value: UpcEanType.SupplementalModeType): SettingCommand
```

```
10.
11.     @JvmStatic //E4770 미지원
12.     fun setSupplementalAimIdFormat(value: UpcEanType.SupplementalAimIdFormatType):
    SettingCommand
13.
14.     @JvmStatic //E4770 미지원
15.     fun setReducedQuietZone(enable: Boolean): SettingCommand
16.
17.     @JvmStatic //E4770 미지원
18.     fun setBookland(value: UpcEanType.BooklandType): SettingCommand
19.
20.     @JvmStatic //E4770 미지원
21.     fun setUccCouponExtend(enable: Boolean): SettingCommand
22.
23.     @JvmStatic //E4770 미지원
24.     fun setCouponReport(value: UpcEanType.CouponReportType): SettingCommand
25.
26.     @JvmStatic //E4770 미지원
27.     fun setIssnEan(enable: Boolean): SettingCommand
28.
29.     @JvmStatic //E4770 미지원
30.     fun setTranslateUpcAToEan13(enable: Boolean): SettingCommand
31.
32.     @JvmStatic //E4770 미지원
33.     fun setSupplementalRedundancy(value: Int): SettingCommand
34.
35.     @JvmStatic //E4770 미지원
36.     fun setEanSupplemental2(enable: Boolean): SettingCommand
37.
38.     @JvmStatic //E4770 미지원
39.     fun setEanSupplemental5(enable: Boolean): SettingCommand
40.
41.     @JvmStatic //SE4107, SE5500 미지원
42.     fun setEan8ReadSupplementals(enable: Boolean): SettingCommand
43.
44.     @JvmStatic //SE4107, SE5500 미지원
45.     fun setEan13ReadSupplementals(enable: Boolean): SettingCommand
46.
47.     @JvmStatic //SE4107, SE5500 미지원
48.     fun setEan8Supplemental2(enable: Boolean): SettingCommand
49.
50.     @JvmStatic //SE4107, SE5500 미지원
51.     fun setEan13Supplemental2(enable: Boolean): SettingCommand
52.
53.     @JvmStatic //SE4107, SE5500 미지원
54.     fun setEan8Supplemental5(enable: Boolean): SettingCommand
55.
56.     @JvmStatic //SE4107, SE5500 미지원
57.     fun setEan13Supplemental5(enable: Boolean): SettingCommand
58. }
```

- setEan8Supplemental2(enable), setEan8Supplemental5(enable) 는 **setEan8ReadSupplementals(true)가 선행**되어야 동작합니다.

- setEan13Supplemental2(enable), setEan13Supplemental5(enable) 는 **setEan13ReadSupplementals(true)가 선행**되어야 동작합니다.

10.35.1 UpcEanType

```
object UpcEanType {
    enum class SupplementalModeType : TypeValue {
        IGNORE_SUPPLEMENTALS,
        DECODE_WITH_SUPPLEMENTALS,
        AUTODISCRIMINATE_SUPPLEMENTALS,
        SMART_SUPPLEMENTAL_MODE,
        ENABLE_378_379,
        ENABLE_978_979,
        ENABLE_414_419_434_439,
        ENABLE_977,
        ENABLE_491,
        USER_PROGRAMMABLE_TYPE1,
        USER_PROGRAMMABLE_TYPE1_2,
        SMART_PLUS_USER_PROGRAMMABLE1,
        SMART_PLUS_USER_PROGRAMMABLE1_2
    }

    enum class SupplementalAimIdFormatType : TypeValue {
        SEPARATE, COMBINED, SEPARATE_TRANSMISSION
    }

    enum class CouponReportType : TypeValue {
        OLD_FORMAT, NEW_FORMAT, AUTODISCRIMINATE
    }

    enum class BooklandType : TypeValue {
        ISBN10, ISBN13
    }
}
```

10.36 UPCE

```
object UPCE {
    @JvmStatic
    fun setEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setConvertToUpcA(enable: Boolean): SettingCommand

    @JvmStatic
    fun setPreamble(value: UpcEType.PreambleType): SettingCommand

    @JvmStatic
    fun setReportCheckDigit(enable: Boolean): SettingCommand

    @JvmStatic
    fun setSupplemental2Bit(enable: Boolean): SettingCommand
```

```
16.
17.     @JvmStatic
18.     fun setSupplemental5Bit(enable: Boolean): SettingCommand
19. }
```

10.36.1 UpcEType

```
1.  object UpcEType {
2.      enum class PreambleType : TypeValue {
3.          NONE, SYS_CHAR, COUNTRY_SYS_CHAR
4.      }
5.  }
```

10.37 UpuFicsPostal

```
1.  object UpuFicsPostal {
2.      @JvmStatic //E4770 미지원
3.      fun setEnable(enable: Boolean): SettingCommand
4.  }
```

10.38 USPlanet

```
1.  object USPlanet {
2.      @JvmStatic //E4770 미지원
3.      fun setEnable(enable: Boolean): SettingCommand
4.
5.      @JvmStatic //E4770 미지원
6.      fun setReportCheckDigit(enable: Boolean): SettingCommand
7.  }
```

10.39 USPostnet

```
1.  object USPostnet {
2.      @JvmStatic //E4770 미지원
3.      fun setEnable(enable: Boolean): SettingCommand
4.  }
```

10.40 General

```
1.  object General {
2.      @JvmStatic
3.      fun findMe(): SettingCommand
4.
5.      @JvmStatic
```

```
    fun setAimer(enable: Boolean): SettingCommand

    @JvmStatic
    fun setIllumination(enable: Boolean): SettingCommand

    @JvmStatic
    fun setSound(type: SoundType): SettingCommand

    @JvmStatic
    fun setVibrate(enable: Boolean): SettingCommand

    @JvmStatic
    fun setLedEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setButtonEnable(enable: Boolean): SettingCommand

    @JvmStatic
    fun setWithCodeType(enable: Boolean): SettingCommand
}
```

10.40.1 SoundType

```
enum class SoundType : TypeValue {
    ON, OFF, MUTE_ON_FAILED_SCAN
}
```

10.41 ReaderParams

```
object ReaderParams {
    @JvmStatic
    fun setReadMode(value: ReaderType.ReadModeType): SettingCommand

    /** value is in deciseconds (seconds × 10). e.g., 1s -> 10, 1.5s -> 15, 10s -> 100. */
    @JvmStatic
    fun setLaserOnTime(value: Int): SettingCommand

    @JvmStatic
    fun setInverse1D(value: ReaderType.Inverse1DType): SettingCommand

    @JvmStatic
    fun setQuietZoneLevelFor1D(value: ReaderType.QuietZoneLevelType): SettingCommand

    @JvmStatic
    fun setPoorQualityDecodeEffort(value: ReaderType.PoorQualityDecodeEffortType):
                                                                        SettingCommand
}
```

10.41.1 ReaderType

```
object ReaderType {
    enum class Inverse1DType : TypeValue {
        REGULAR_ONLY, INVERSE_ONLY, INVERSE_AUTODETECT
    }
```

```
5.
6.      enum class QuietZoneLevelType : TypeValue {
7.          LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
8.      }
9.
10.     enum class PoorQualityDecodeEffortType : TypeValue {
11.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
12.     }
13.
14.     enum class ReadModeType : TypeValue {
15.         ASYNC, SYNC
16.     }
17. }
```

10.42 BasicDataFormat

```
1.  object BasicDataFormat {
2.      @JvmStatic
3.      fun setTransmitCodeID(value: BasicFormatType.TransmitCodeIdType): SettingCommand
4.
5.      @JvmStatic
6.      fun setEndChar(value: BasicFormatType.EndCharType): SettingCommand
7.
8.      @JvmStatic
9.      fun setSubString(value: String): SettingCommand
10.
11.     @JvmStatic
12.     fun setRemoveFnc(enable: Boolean): SettingCommand
13.
14.     @JvmStatic
15.     fun setTranslateData(value: String): SettingCommand
16.
17.     @JvmStatic
18.     fun setPrefixPostfixAsAsciiHex(enable: Boolean): SettingCommand
19.
20.     @JvmStatic
21.     fun setPrefix(value: String): SettingCommand
22.
23.     @JvmStatic
24.     fun setPostfix(value: String): SettingCommand
25. }
```

10.42.1 BasicFormatType

```
1.  object BasicFormatType {
2.      enum class TransmitCodeIdType : TypeValue {
3.          NONE, AIM, SYMBOL
4.      }
5.
6.      enum class EndCharType : TypeValue {
7.          ENTER, TAB, NONE
8.      }
9.  }
```

10.42.2 String Value Format

모든 String 의 값은 아래의 형식에 맞춰 넣어주셔야 합니다. 특수기호도 포함해서 넣어주세요.

- setSubstring(String)

설정된 인덱스에 맞춰 스캐너가 출력하는 데이터가 잘려서 들어옵니다.

입력 형식: 시작 인덱스,끝 인덱스

ex) 3,5

- setTranslateData(String)

스캐너가 출력하는 바이트를 원본 바이트 → 치환 바이트로 매핑해 변환합니다.

입력 형식: [원본 값,치환 값];[원본 값,치환 값]... 최대 8 쌍까지

ex) 5B,28;5D,29

- setPrefix(String)

설정한 바이트가 스캐너 출력 데이터 앞에 프리픽스로 붙습니다.

입력 형식: 최대 16 자(16bytes)

Ex) abcdef

- setPostfix(String)

설정한 바이트가 스캐너 출력 데이터 뒤에 프리픽스로 붙습니다.

입력 형식: 최대 16 자(16bytes)

Ex) abcdef

10.43 Dev

```
1. object Dev {
2.     @JvmStatic
3.     fun setBatteryCallback(enable: Boolean): SettingCommand
4. }
```

11. M3Utils

WR15 SDK 를 이용해 앱을 개발하실 때 유용한 함수들입니다.

1. toUtf8String

```
/* M3Utils */

@JvmOverloads
@JvmStatic
fun toUtf8String(bytes: ByteArray, charset: Charset = Charsets.UTF_8): String
```

바이트 배열을 지정한 문자셋으로 디코딩하여 String 으로 변환합니다. 기본값은 UTF-8 입니다.

2. isDecimalPair

```
/* M3Utils */

@JvmStatic
fun isDecimalPair(input: String): Boolean
```

입력이 "Int,Int" 형식의 String 인지 검증합니다.
Settings.BasicDataFormat.setSubString(String)를 호출하실 때, 파라미터의 유효성 검증용으로 사용하실 수 있습니다.

3. sHexPairsSequenceAndMaxLength8

```
/* M3Utils */

@JvmStatic
fun isHexPairsSequenceAndMaxLength8(value: String): Boolean
```

입력이 "hh,hh;hh,hh" 형식의 String 이면서 ;를 기준으로 최대 8 쌍인지 검증합니다.
Setting.BasicDataFormat.setTranslateData(String)를 호출하실 때, 파라미터의 유효성 검증용으로 사용하실 수 있습니다.

4. generateQrBitmap(DeviceState, Int)

```
/* M3Utils */

@JvmOverloads
@JvmStatic
fun generateQrBitmap(deviceState: DeviceState, size: Int = 512): Bitmap?
```

DeviceState 를 이용해 WR15 설정용 QR 코드를 생성합니다.

- deviceState: DeviceState : Qr 로 생성할 객체
- size: Int : Qr 코드의 크기 (기본값 512)

12. ERROR CODE

실패에 맞는 다양한 에러코드를 제공합니다.

12-1 TransportErrorCatalog

```
enum class TransportErrorCatalog(val code: Int, val msg: String) {
    INVALID_COMMAND(1001, "Invalid command"),
    CONNECTION_FAILED(1002, "Connection failed"),
    UNSUPPORTED_SETTING(1003, "Unsupported setting"),
    SDK_ERROR(1004, "SDK internal error"),
    SETTING_FAILED(1005, "Setting request failed"),
    REQUEST_TIMEOUT(1006, "Request timed out"),
    UNKNOWN_ERROR(1100, "Unknown error occurred"),
}
```

12-2 ConnectErrorCatalog

```
enum class ConnectErrorCatalog(val code: Int, val msg: String) {
    ALREADY_CONNECTING(2001, "Already connecting to a device."),
    CONNECTION_FAILED(2004, "Connection failed."),
    DEVICE_SETTINGS_LOAD_FAILED(2005, "Failed to load Device Settings"),
    UNKNOWN(2100, "Unknown connection error"),
}
```