

# M3 MOBILE

## WR15 SDK 메뉴얼



# 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	

# Index

1. Import SDK .....	- 7 -
1.1 Libraries .....	- 7 -
2. Permissions Required .....	- 8 -
2.1 Android 12+(API 31+)이 중심인 경우 .....	- 8 -
2.2 Android 6~11(API25~30)도 지원해야 하는 경우 .....	- 8 -
2.3 런타임 권한 요청 확인 필수 .....	- 8 -
3. Initialize SDK .....	- 9 -
3.1 Initialize the dependence libraries .....	- 9 -
3.1.1 M3Wr15SdkConfig .....	- 9 -
3.1.2 Context .....	- 9 -
4. Search for WR15 .....	- 10 -
4.1 S2P (Scan to Pair) .....	- 10 -
4.1.1 Generate Barcode .....	- 10 -
4.1.2 Start S2P Scan .....	- 10 -
4.1.3 Stop S2P Scan .....	- 11 -
4.2 Search and Connect .....	- 11 -
4.2.1 Start Discovery Scan .....	- 11 -
4.2.2 Stop Discovery Scan .....	- 12 -
5. Connect to WR15 .....	- 13 -
5.1 Connect .....	- 13 -
6. Disconnect .....	- 14 -
6.1 Disconnect to WR15 .....	- 14 -
6.2 Observe Disconnect .....	- 14 -
7. Receive Barcode Data .....	- 16 -
7.1 Add Barcode Data .....	- 16 -

7.2 Remove Barcode Data.....	- 16 -
8. Scanner Device Information.....	- 17 -
8.1 Observe DeviceState .....	- 17 -
8.2 Set Scanner Setting .....	- 17 -
8.2.1 SettingCommand.....	- 18 -
8.2.2 Generate SettingCommand.....	- 18 -
8.2.3 Caution.....	- 19 -
9. DeviceState .....	- 20 -
9.1 Strucutre .....	- 20 -
9.1.1 DeviceInfo .....	- 20 -
9.1.2 GeneralSettings.....	- 20 -
9.1.3 Symbologies.....	- 20 -
9.1.4 BasicFormat .....	- 30 -
9.1.5 readerSettings .....	- 30 -
9.1.6 DevSettings.....	- 30 -
10. Settings .....	- 31 -
10.1 AustralianPostal.....	- 31 -
10.2 Aztec .....	- 31 -
10.3 Chinese2of5 .....	- 31 -
10.4 Codabar .....	- 31 -
10.4.1 CodabarType .....	- 32 -
10.5 Code11 .....	- 32 -
10.5.1 Code11Type.....	- 32 -
10.6 Code39.....	- 32 -
10.6.1 Code39Type.....	- 33 -
10.7 Code93.....	- 33 -
10.8 Code128 .....	- 33 -

10.8.1 Code128Type .....	- 34 -
10.9 Composite .....	- 34 -
10.9.1 CompositeType .....	- 35 -
10.10 DataMatrix.....	- 35 -
10.10.1 DatatMatrixType .....	- 35 -
10.11 Discrete2Of5 .....	- 35 -
10.12 DotCode.....	- 36 -
10.12.1 DotCodeType.....	- 36 -
10.13 EAN8.....	- 36 -
10.14 EAN13 .....	- 36 -
10.15 Gs1DataBar14 .....	- 37 -
10.16 Gs1DataBarExpanded .....	- 37 -
10.17 Gs1DataBarLimited .....	- 37 -
10.17.1 Gs1DataBarLimitedType .....	- 37 -
10.18 Gs1128.....	- 37 -
10.19 HanXin .....	- 38 -
10.20 Interleaved2Of5 .....	- 38 -
10.20.1 Interleaved2Of5Type .....	- 38 -
10.21 ISBT128.....	- 39 -
10.22 JapanesePostal .....	- 39 -
10.23 Korean3of5 .....	- 39 -
10.24 Matrix2of5.....	- 39 -
10.25 MaxiCode .....	- 39 -
10.26 MicroPdf417.....	- 41 -
10.27 MicroQrCode .....	- 41 -
10.28 MSI .....	- 41 -
10.28.1 MsiType.....	- 41 -

10.29 NetherlandsKix .....	- 42 -
10.30 Pdf417 .....	- 42 -
10.31 QrCode .....	- 42 -
10.32 UKPostal .....	- 42 -
10.33 UPCA .....	- 42 -
10.33.1 UpcAType .....	- 43 -
10.34 UPCE1 .....	- 43 -
10.34.1 UpcE1Type .....	- 43 -
10.35 UPCEan .....	- 43 -
10.35.1 UpcEanType .....	- 45 -
10.36 UPCE .....	- 45 -
10.36.1 UpcEType .....	- 46 -
10.37 UpuFicsPostal .....	- 46 -
10.38 USPlanet .....	- 46 -
10.39 USPostnet .....	- 46 -
10.40 General .....	- 46 -
10.40.1 SoundType .....	- 47 -
10.41 ReaderParams .....	- 47 -
10.41.1 ReaderType .....	- 47 -
10.42 BasicDataFormat .....	- 48 -
10.42.1 BasicFormatType .....	- 48 -
10.42.2 String Value Format .....	- 49 -
10.43 Dev .....	- 49 -
11. M3Utils .....	- 50 -

# 1. Import SDK

## 1.1 Libraries

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

```
1. dependencyResolutionManagement {  
2.     repositories {  
3.         google()  
4.         mavenCentral()  
5.  
6.         //이 줄 추가  
7.         maven("https://m3mobile.github.io/wr15-sdk-maven")  
8.     }  
9. }
```

\* **Groovy** 를 사용하시는 경우

```
1. dependencyResolutionManagement {  
2.     repositories {  
3.         google()  
4.         mavenCentral()  
5.  
6.         //이 줄 추가  
7.         maven { url "https://m3mobile.github.io/wr15-sdk-maven" }  
8.     }  
9. }  
10. }
```

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

```
1. dependencies {  
2.     implementation("com.m3:wr15-sdk:1.2.0") //원하시는 버전을 추가해 주세요.  
3.     ...  
4. }
```

## 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

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

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

1. `<!-- Android 6~11 (API 25~30) -->`
2. `<uses-permission android:name="android.permission.BLUETOOTH"`
3. `android:maxSdkVersion="30"/>`
4. `<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"`
5. `android:maxSdkVersion="30"/>`
6. `<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"`
7. `android:maxSdkVersion="30"/>`
8. `<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"`
9. `android:maxSdkVersion="30"/>`
10. `<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"`
11. `android:maxSdkVersion="30"/>`
- 12.
13. `<!-- Android 12+ (API 31+) -->`
14. `<uses-permission android:name="android.permission.BLUETOOTH_SCAN"`
15. `android:usesPermissionFlags="neverForLocation"/>`
16. `<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

```
1. val config = M3Wr15SdkConfig.builder()
2.     .connectionType(ConnectionType.SPP)
3.     .enableLog(true)
4.     .build()
5.
6.
7. M3Wr15Sdk.initialize(context, config)
8.
```

#### Java

```
1. M3Wr15SdkConfig config = M3Wr15SdkConfig.builder()
2.     .connectionType(ConnectionType.SPP)
3.     .enableLog(true)
4.     .build();
5.
6. M3Wr15Sdk sdk =
7.     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 SharedPreferences에 사용되기 때문에, **Activity Context**가 아닌 **Application Context**를 전달하세요.

## 4. Search for WR15

### 4.1 S2P (Scan to Pair)

WR15 장비로 S2P 바코드를 스캔하여, 스캔한 장비를 검색합니다.

#### 4.1.1 Generate Barcode

```
1. /* M3Wr15Sdk */
2.
3. @JvmOverloads
4. @JvmStatic
5. fun getBarcodeBitmapForS2P(width: Int = 260, height: Int = 50): Bitmap?
6.
7. @JvmStatic
8. 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

```
1. /* M3Wr15Sdk */
2.
3. @RequiresPermission(
4.     allOf = [Manifest.permission.BLUETOOTH_SCAN, Manifest.permission.BLUETOOTH_CONNECT])
5. fun startS2PScan(
6.     onDeviceFound: (BluetoothDevice) -> Unit,
7.     onScanFailed: (errorCode: Int) -> Unit
8. )
```

##### For Java

```
1. /* M3Wr15Sdk */
2.
3. @JvmStatic
4. @RequiresPermission(
5.     allOf = [Manifest.permission.BLUETOOTH_SCAN, Manifest.permission.BLUETOOTH_CONNECT])
6. fun startS2PScan(listener: S2PScanResultListener)
7.
```

```

8. interface S2PScanResultListener {
9.     fun onDeviceFound(device: BluetoothDevice)
10.    fun onScanFailed(errorCode: Int)
11. }

```

*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 *startS2PScan()*을 중단하는 것이 바람직합니다.

```

1.  /* M3Wr15Sdk */
2.
3.  @RequiresPermission(Manifest.permission.BLUETOOTH_SCAN)
4.  fun stopS2PScan()

```

#### Example

```

1.  @RequiresPermission(Manifest.permission.BLUETOOTH_SCAN)
2.  fun stopS2PScan()
3.
4.  M3Wr15Sdk.startS2PScan(object: S2PScanResultListener) {
5.      override fun onDeviceFound(device: BluetoothDevice) {
6.          M3Wr15Sdk.stopS2PScan() //스캔 중단
7.          // 발견한 BluetoothDevice 로 연결 시도
8.      }
9.
10.     override fun onScanFailed(errorCode: Int) {
11.         // 스캔 실패 처리
12.     }
13. })

```

## 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.

```

1.  /*M3Wr15Sdk*/
2.
3.  @RequiresPermission(allOf = [Manifest.permission.BLUETOOTH_SCAN,
4.      Manifest.permission.BLUETOOTH_CONNECT])
5.  fun startDiscoveryScan(listener: DiscoveryScanResultListener)

```

```

6. interface DiscoveryScanResultListener {
7.     fun onStart() //스캔 시작
8.     fun onError() // 에러 발생
9.     fun onDeviceFound(device: DiscoveryDeviceModel) //WR15 스캐너 장치 발견
10.    fun onFinish() //스캔 종료
11. }

```

### Exmaple

```

1. M3Wr15Sdk.startDiscoveryScan(object : DiscoveryScanResultListener {
2.     @RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
3.     override fun onDeviceFound(device: DiscoveryDeviceModel) {
4.         // 디바이스 발견 처리
5.     }
6.
7.     override fun onError(errorMsg: String) {
8.         // 에러 처리
9.     }
10.
11.    override fun onFinish() {
12.        // 스캔 완료 처리
13.    }
14.
15.    override fun onStart() {
16.        // 스캔 시작 처리
17.    }
18. }
19. )

```

## 4.2.2 Stop Discovery Scan

Discovery Scan 을 중지합니다.

```

1. /*M3Wr15Sdk*/
2.
3. @JvmStatic
4. fun stopDiscoveryScan()

```

## 5. Connect to WR15

### 5.1 Connect

BluetoothDevice 장비를 SDK 와 연결합니다.

#### For Kotlin

```

1. /*M3Wr15Sdk*/
2.
3. @RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
4. fun connectToDevice(
5.     device: BluetoothDevice,
6.     onSuccess: (session: TransportSession) -> Unit,
7.     onFailure: (errorMsg: String) -> Unit
8. )

```

#### For Java

```

1. /*M3Wr15Sdk*/
2.
3. @JvmStatic
4. @RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
5. fun connectToDevice(device: BluetoothDevice, listener: ConnectResultListener)
6.
7. interface ConnectResultListener {
8.     fun onSuccess(session: TransportSession)
9.     fun onFailure(errorMsg: String)
10. }

```

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

#### Example

```

1. val sdk = M3Wr15Sdk.initialize(applicationContext, config)
2.
3. sdk.startS2PScan(object: DeviceFindResultListener {
4.     override fun onDeviceFound(device: BluetoothDevice) {
5.         sdk.stopS2PScan()
6.         sdk.connectToDevice(device, object:ConnectResultListener {
7.             override fun onSuccess(session: TransportSession) {
8.                 //성공 처리
9.             }
10.            override fun onFailure(errorMsg: String) {
11.                //실패 처리
12.            }
13.        })
14.    }
15.
16.    override fun onScanFailed(errorCode: Int) {
17.        // 스캔 실패 처리

```

```
18.     }
19. })
```

## 6. Disconnect

### 6.1 Disconnect to WR15

연결되어 있는 WR15 스캐너와의 연결을 해제합니다.

```
1. /* TransportSession */
2.
3. //For Kotlin
4. fun disconnect(onSuccess: () -> Unit, onFailure: (errorMsg: String) -> Unit)
5.
6. //For Java
7. fun disconnect(listener: DisconnectResultListener)
```

### 6.2 Observe Disconnect

연결이 해제되는 순간을 감지합니다.

```
1. /* TransportSession */
2.
3. fun addDisconnectListener(disconnectListener: DisconnectListener)
4. fun removeDisconnectListener(disconnectListener: DisconnectListener)
5.
6. fun interface DisconnectListener {
7.     fun onDisconnect()
8. }
```

#### Example

```
1. val disconnectListener = object: DisconnectListener {
2.     override fun onDisconnect() {
3.         // 연결 해제 시 처리
4.     }
5. }
6. fun connectToDevice() {
7.     sdk.connectToDevice(device, object:ConnectResultListener {
8.         override fun onSuccess(session:TransportSession) {
9.             session.addDisconnectListener(disconnectListener)
10.        }
11.        override fun onFailure(errorMsg: String) {
12.            //실패 처리
13.        }
14.    })
15. }
```

```
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.

```
1. /* TransportSession */
2.
3. fun addDecodingDataListener(decodingDataListener: DecodingDataListener)
4.
5. interface DecodingDataListener {
6.     fun onReceive(decodingData: ByteArray)
7. }
```

### 7.2 Remove Barcode Data

BarcodeDataListener 를 해제합니다.

```
1. /* TransportSession */
2.
3. 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.

```

1. /* TransportSession */
2.
3. fun addDeviceStateListener(listener: DeviceStateListener)
4. fun removeDeviceStateListener(listener: DeviceStateListener)
5.
6. interface DeviceStateListener {
7.     fun onDeviceStateChanged(state: DeviceState)
8. }
9.
10. data class DeviceState(
11.     val deviceInfo: DeviceInfo = DeviceInfo(),
12.     val general: GeneralSettings = GeneralSettings(),
13.     val symbolologies: Symbolologies = Symbolologies(),
14.     val basicFormat: BasicFormat = BasicFormat(),
15.     val readerSettings: ReaderSettings = ReaderSettings(),
16.     val devSettings: DevSettings = DevSettings()
17. )

```

### 8.2 Set Scanner Setting

WR15 스캐너의 설정을 변경합니다.

#### For Kotlin

```

1. /* TransportSession */
2.
3. fun setSetting(setting: SettingCommand, onSuccess: () -> Unit, onFailure: (errorMsg:
    String) -> Unit)

```

#### For Java

```

1. /* TransportSession */
2.
3. fun setSetting(setting: SettingCommand, listener: SetSettingResultListener)
4.
5. interface SetSettingResultListener{
6.     fun onSuccess()
7.     fun onFailure(errorMsg: String)
8. }

```

### 8.2.1 SettingCommand

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

### 8.2.2 Generate SettingCommand

*Settings* 는 *SettingsCommand* 를 손쉽게 생성하도록 돕는 API 진입점입니다.

```

1. object Settings {
2.     object Codabar {
3.         @JvmStatic
4.         fun setEnable(enable: Boolean): SettingCommand
5.         @JvmStatic
6.         fun setLength1(length: Int): SettingCommand
7.         ...
8.     }
9.
10.    object Code11 {
11.        @JvmStatic
12.        fun setEnable(enable: Boolean): SettingCommand
13.        @JvmStatic
14.        fun setLength1(length: Int): SettingCommand
15.        ...
16.    }
17.
18.    object General {
19.        @JvmStatic
20.        fun findMe(): SettingCommand
21.        @JvmStatic
22.        fun setAimer(enable: Boolean): SettingCommand
23.        @JvmStatic
24.        fun setIllumination(enable: Boolean): SettingCommand
25.        @JvmStatic
26.        fun setSound(type: SoundType): SettingCommand
27.        @JvmStatic
28.        fun setVibrate(enable: Boolean): SettingCommand
29.        ...
30.    }
31.    ... //다른 설정들
32. }
```

더 자세한 설정 항목은 [10.Settings](#) 를 참조해 주세요.

## Example

### For Kotlin

```
1. session.setSetting(  
2.     Settings.EAN13.setEnable(true),  
3.     onSuccess = {  
4.         // 설정 성공 처리  
5.     },  
6.     onFailure = { errorMsg ->  
7.         // 설정 실패 처리  
8.     }  
9. )
```

### For Java

```
1. session.setSetting(  
2.     Settings.General.setSound(SoundType.OFF),  
3.     object : SetSettingResultListener {  
4.         override fun onFailure(errorMsg: String) {  
5.             // 설정 실패 처리  
6.         }  
7.  
8.         override fun onSuccess() {  
9.             // 설정 성공 처리  
10.        }  
11.    }  
12. )
```

### 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

```
1. data class DeviceState(  
2.     val deviceInfo: DeviceInfo = DeviceInfo(),  
3.     val general: GeneralSettings = GeneralSettings(),  
4.     val symbolologies: Symbolologies = Symbolologies(),  
5.     val basicFormat: BasicFormat = BasicFormat(),  
6.     val readerSettings: ReaderSettings = ReaderSettings(),  
7.     val devSettings: DevSettings = DevSettings()  
8. )
```

#### 9.1.1 DeviceInfo

```
1. data class DeviceInfo(  
2.     val name: String = "",  
3.     val macAddress: String = "",  
4.     val model: String = "",  
5.     val vendor: VendorType? = null,  
6.     val serialNumber: String = "",  
7.     val fwVersion: String = "",  
8.     val releaseDate: String = "",  
9.     val batteryPercent: Int? = null,  
10.    val scannerId: ScannerIdType? = null,  
11.    val btMode: BtModeType? = null,  
12. )
```

#### 9.1.2 GeneralSettings

```
1. data class GeneralSettings(  
2.     val sound: SoundType? = null,  
3.     val vibrateEnabled: Boolean? = null,  
4.     val ledEnabled: Boolean? = null,  
5.     val buttonEnabled: Boolean? = null,  
6.     val aimEnabled: Boolean? = null,  
7.     val illuminationEnabled: Boolean? = null,  
8. )
```

#### 9.1.3 Symbolologies

```
1. data class Symbolologies(  
2.     val australianPostalVals: AustralianVals = AustralianVals(),  
3.     val aztecVals: AztecVals = AztecVals(),  
4.     val chinese20f5Vals: Chinese20f5Vals = Chinese20f5Vals(),
```

```

5.    val codabarVals: CodabarVals = CodabarVals(),
6.    val code11Vals: Code11Vals = Code11Vals(),
7.    val code39Vals: Code39Vals = Code39Vals(),
8.    val code93Vals: Code93Vals = Code93Vals(),
9.    val code128Vals: Code128Vals = Code128Vals(),
10.   val compositeVals: CompositeVals = CompositeVals(),
11.   val dataMatrixVals: DataMatrixVals = DataMatrixVals(),
12.   val discrete20f5Vals: Discrete20f5Vals = Discrete20f5Vals(),
13.   val dotCodeVals: DotCodeVals = DotCodeVals(),
14.   val ean8Vals: Ean8Vals = Ean8Vals(),
15.   val ean13Vals: Ean13Vals = Ean13Vals(),
16.   val gs1DataBar14Vals: Gs1DataBar14Vals = Gs1DataBar14Vals(),
17.   val gs1DataBarExpandedVals: Gs1DataBarExpandedVals = Gs1DataBarExpandedVals(),
18.   val gs1DataBarLimitedVals: Gs1DataBarLimitedVals = Gs1DataBarLimitedVals(),
19.   val gs1128Vals: Gs1128Vals = Gs1128Vals(),
20.   val hanXinVals: HanXinVals = HanXinVals(),
21.   val interleaved20f5Vals: Interleaved20f5Vals = Interleaved20f5Vals(),
22.   val isbt128Vals: Isbt128Vals = Isbt128Vals(),
23.   val japanesePostalVals: JapanesePostalVals = JapanesePostalVals(),
24.   val korean30f5Vals: Korean30f5Vals = Korean30f5Vals(),
25.   val matrix20f5Vals: Matrix20f5Vals = Matrix20f5Vals(),
26.   val maxiCodeVals: MaxiCodeVals = MaxiCodeVals(),
27.   val microPdf417Vals: MicroPdf417Vals = MicroPdf417Vals(),
28.   val microQrCodeVals: MicroQrCodeVals = MicroQrCodeVals(),
29.   val msiVals: MsiVals = MsiVals(),
30.   val netherlandsKixVals: NetherlandsKixVals = NetherlandsKixVals(),
31.   val pdf417Vals: Pdf417Vals = Pdf417Vals(),
32.   val qrCodeVals: QrCodeVals = QrCodeVals(),
33.   val ukPostalVals: UkPostalVals = UkPostalVals(),
34.   val upcAVals: UpcAVals = UpcAVals(),
35.   val upcE1Vals: UpcE1Vals = UpcE1Vals(),
36.   val upcEanVals: UpcEanVals = UpcEanVals(),
37.   val upcEVals: UpcEVals = UpcEVals(),
38.   val upuFicsPostalVals: UpuFicsPostalVals = UpuFicsPostalVals(),
39.   val usPlanetVals: UsPlanetVals = UsPlanetVals(),
40.   val usPostnetVals: UsPostnetVals = UsPostnetVals(),
41. )

```

### AustralianVals

지원하는 Scanner : *SE4107, SE5500*

```

1. data class AustralianVals(
2.     val enabled: Boolean? = null,
3. )

```

### AztecVals

지원하는 Scanner : *SE4107, SE5500, E4770*

```

1. data class AztecVals(
2.     val enabled: Boolean? = null
3. )

```

### Chinese20f5Vals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class Chinese20f5Vals(  
2.     val enabled: Boolean? = null  
3. )
```

### CodabarVals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class CodabarVals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null,  
5.     val clsiEditing: Boolean? = null, //E4770 미지원  
6.     val notisEditing: Boolean? = null,  
7.     val securityLevel: CodabarType.SecurityType? = null, //E4770 미지원  
8. )
```

### Code11Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class Code11Vals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null,  
5.     val reportCheckDigit: Boolean? = null,  
6.     val verifyCheckDigit: Code11Type.VerifyCheckDigitType? = null  
7. )
```

### Code39Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class Code39Vals(  
2.     val enabled: Boolean? = null,  
3.     val triopticCode39: Boolean? = null, //E4770 미지원  
4.     val length1: Int? = null,  
5.     val length2: Int? = null,  
6.     val reducedQuietZone: Boolean? = null, //E4770 미지원  
7.     val convertToCode32: Boolean? = null,  
8.     val fullAscii: Boolean? = null,  
9.     val reportCheckDigit: Boolean? = null, // true = TRANSMIT  
10.    val reportCode32Prefix: Boolean? = null,  
11.    val securityLevel: Code39Type.SecurityType? = null, //E4770 미지원  
12.    val verifyCheckDigit: Boolean? = null // true = ENABLE  
13. )
```

### Code93Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class Code93Vals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,
```

```

4.     val length2: Int? = null
5. )

```

### Code128Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```

1. data class Code128Vals(
2.     val enabled: Boolean? = null,
3.     val length1: Int? = null,
4.     val length2: Int? = null,
5.     val reducedQuietZone: Boolean? = null, //E4770 미지원
6.     val ignoreFnc4: Boolean? = null, //E4770 미지원
7.     val checkIsbtTable: Boolean? = null, //E4770 미지원
8.     val isbt128ConcatMode: Code128Type.ISBT128ConcatModeType? = null, //E4770 미지원
9.     val securityLevel: Code128Type.SecurityType? = null, //E4770 미지원
10.    val emulationMode: Boolean? = null //E4770 미지원
11. )

```

### CompositeVals

지원하는 Scanner : *SE4107, SE5500, E4770*

```

1. data class CompositeVals(
2.     val enabled: Boolean? = null, //SE4107, SE5500 미지원
3.     val compositeABEnabled: Boolean? = null, //E4770 미지원
4.     val compositeCEnabled: Boolean? = null, //E4770 미지원
5.     val tlc39Enable: Boolean? = null, //E4770 미지원
6.     val upcCompositeMode: CompositeType.UPCCompositeModeType? = null //E4770 미지원
7. )

```

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*

```

1. data class DataMatrixVals(
2.     val enabled: Boolean? = null,
3.     val inverse: DataMatrixType.InverseType? = null,
4.     val decodeMirrorImages: DataMatrixType.DecodeMirrorImagesType? = null, //E4770 미지원
5.     val gs1DataMatrix: Boolean? = null //E4770 미지원
6. )

```

### Discrete2Of5Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```

1. data class Discrete2Of5Vals(
2.     val enabled: Boolean? = null,

```

```
3.     val length1: Int? = null,  
4.     val length2: Int? = null  
5. )
```

### DotCodeVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class DotCodeVals(  
2.     val enabled: Boolean? = null,  
3.     val inverse: DotCodeType.InverseType? = null,  
4.     val mirror: DotCodeType.MirrorType? = null,  
5.     val prioritize: Boolean? = null  
6. )
```

### Ean8Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class Ean8Vals(  
2.     val enabled: Boolean? = null  
3. )
```

### Ean13Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class Ean13Vals(  
2.     val enabled: Boolean? = null  
3. )
```

### Gs1DataBar14Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class Gs1DataBar14Vals(  
2.     val enabled: Boolean? = null  
3. )
```

### Gs1DataBarExpandedVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class Gs1DataBarExpandedVals(  
2.     val enabled: Boolean? = null  
3. )
```

### Gs1DataBarLimitedVals

지원하는 Scanner : *SE4107, SE5500*

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



```

3.     val securityLevel: Gs1DataBarLimitedType.SecurityType? = null
4. )

```

### Gs1128Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```

1. data class Gs1128Vals(
2.     val enabled: Boolean? = null
3. )

```

### HanXinVals

지원하는 Scanner : *SE4107, SE5500, E4770*

```

1. data class HanXinVals(
2.     val enabled: Boolean? = null,
3.     val inverse: HanXinType.InverseType? = null
4. )

```

### Interleaved2Of5Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```

1. data class Interleaved2Of5Vals(
2.     val enabled: Boolean? = null,
3.     val length1: Int? = null,
4.     val length2: Int? = null,
5.     val checkDigit: Interleaved2Of5Type.CheckDigitType? = null,
6.     val reportCheckDigit: Boolean? = null,    // true = TRANSMIT
7.     val securityLevel: Interleaved2Of5Type.SecurityType? = null, //E4770 미지원
8.     val convertItf14ToEan13: Boolean? = null, //E4770 미지원
9.     val reducedQuietZone: Boolean? = null    //E4770 미지원
10. )

```

### Isbt128Vals

지원하는 Scanner : *SE4107, SE5500*

```

1. data class Isbt128Vals(
2.     val enabled: Boolean? = null
3. )

```

### JapanesePostalVals

지원하는 Scanner : *SE4107, SE5500*

```

1. data class JapanesePostalVals(
2.     val enabled: Boolean? = null
3. )

```

### Korean3Of5Vals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class Korean3Of5Vals(  
2.     val enabled: Boolean? = null  
3. )
```

### Matrix2Of5Vals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class Matrix2Of5Vals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null,  
5.     val redundancy: Boolean? = null, //E4770 미지원  
6.     val reportCheckDigit: Boolean? = null, // true = TRANSMIT  
7.     val verifyCheckDigit: Boolean? = null // true = ENABLE  
8. )
```

### MaxiCode

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class MaxiCodeVals(  
2.     val enabled: Boolean? = null  
3. )
```

### MicroPdf417Vals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class MicroPdf417Vals(  
2.     val enabled: Boolean? = null  
3. )
```

### MicroQrCodeVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class MicroQrCodeVals(  
2.     val enabled: Boolean? = null  
3. )
```

### MsiVals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class MsiVals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null,  
5.     val checkDigit: MsiType.CheckDigitType? = null,  
6.     val checkDigitScheme: MsiType.CheckDigitSchemeType? = null,  
7.     val reportCheckDigit: Boolean? = null // true = TRANSMIT
```

8. )

### NetherlandKixVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class NetherlandsKixVals(  
2.     val enabled: Boolean? = null  
3. )
```

### Pdf417Vals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class Pdf417Vals(  
2.     val enabled: Boolean? = null  
3. )
```

### QrCodeVals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class QrCodeVals(  
2.     val enabled: Boolean? = null  
3. )
```

### UkPostalVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class UkPostalVals(  
2.     val enabled: Boolean? = null,  
3.     val reportCheckDigit: Boolean? = null // true = TRANSMIT  
4. )
```

### UpcAVals

지원하는 Scanner : *SE4107, SE5500, E4770*

```
1. data class UpcAVals(  
2.     val enabled: Boolean? = null,  
3.     val reportCheckDigit: Boolean? = null, // true = TRANSMIT  
4.     val preamble: UpcAType.PreambleType? = null,  
5.     val upcA2BitsSupplementals: Boolean? = null,  
6.     val upcA5BitsSupplementals: Boolean? = null  
7. )
```

### UpcE1Vals

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

```
1. data class UpcE1Vals(
2.     val enabled: Boolean? = null,
3.     val convertToUpca: Boolean? = null, // true = CONVERT //E4770 미지원
4.     val preamble: UpcE1Type.PreambleType? = null, //E4770 미지원
5.     val reportCheckDigit: Boolean? = null // true = TRANSMIT //E4770 미지원
6. )
```

### UPpcEanVals

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

```
1. data class UpcEanVals(
2.     val reportEan8CheckDigit: Boolean? = null, // true = TRANSMIT
3.     val reportEan13CheckDigit: Boolean? = null, // true = TRANSMIT
4.
5.     val supplementalMode: UpcEanType.SupplementalModeType? = null, //E4770 미지원
6.     val supplementalAimIdFormat: UpcEanType.SupplementalAimIdFormatType? = null, //E4770 미지원
7.     val reducedQuietZone: Boolean? = null, //E4770 미지원
8.     val bookland: UpcEanType.BooklandType? = null, //E4770 미지원
9.     val uccCouponExtend: Boolean? = null, //E4770 미지원
10.    val couponReport: UpcEanType.CouponReportType? = null, //E4770 미지원
11.    val issnEan: Boolean? = null, //E4770 미지원
12.    val translateUpcaToEan13: Boolean? = null, //E4770 미지원
13.    val supplementalRedundancy: Int? = null, //E4770 미지원
14.    val supplemental2: Boolean? = null, //E4770 미지원
15.    val supplemental5: Boolean? = null, //E4770 미지원
16.
17.    val ean8ReadSupplements: Boolean? = null, //SE4107, SE5500 미지원
18.    val ean13ReadSupplements: Boolean? = null, // //SE4107, SE5500 미지원
19.    val ean8Supplemental2: Boolean? = null, // //SE4107, SE5500 미지원
20.    val ean13Supplemental2: Boolean? = null, // //SE4107, SE5500 미지원
21.    val ean8Supplemental5: Boolean? = null, // //SE4107, SE5500 미지원
22.    val ean13Supplemental5: Boolean? = null // //SE4107, SE5500 미지원
23. )
```

SE4107, SE5500 에서 지원하는 *Supplemental2* 의 기능은 E4770 에서 *ean8Supplemental2*, *ean8Supplemental5* 로 나뉘어져 있고, *Supplemental5* 의 기능은 *ean13Supplemental2*, *ean13Supplemental5* 로 나뉘어져 있습니다.

추가적으로, E4770 에서 *ean8Supplemental2*, *ean8Supplemental5* 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**

```
1. data class UpcEVals(
2.     val enabled: Boolean? = null,
3.     val convertToUpca: Boolean? = null, // true = CONVERT
4.     val preamble: UpcEType.PreambleType? = null,
5.     val reportCheckDigit: Boolean? = null, // true = TRANSMIT
6.     val upcE2BitsSupplementals: Boolean? = null,
7.     val upcE5BitsSupplementals: Boolean? = null
```

8. )

### UpuFicsPostalVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class UpuFicsPostalVals(  
2.     val enabled: Boolean? = null  
3. )
```

### UsPlanetVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class UsPlanetVals(  
2.     val enabled: Boolean? = null,  
3.     val reportCheckDigit: Boolean? = null // true = TRANSMIT  
4. )
```

### UsPostnetVals

지원하는 Scanner : *SE4107, SE5500*

```
1. data class UsPostnetVals(  
2.     val enabled: Boolean? = null  
3. )
```

### 9.1.4 BasicFormat

```
1. data class BasicFormat(  
2.     val transmitCodeId: BasicFormatType.TransmitCodeIdType? = null,  
3.     val endChar: BasicFormatType.EndCharType? = null,  
4.     val substringFormation: String = "",  
5.     val removeFnc: Boolean? = null,  
6.     val translateData: String = "",  
7.     val prefixPostfixAsAsciiHex: Boolean? = null,  
8.     val prefix: String = "",  
9.     val postfix: String = ""  
10. )
```

### 9.1.5 readerSettings

```
1. data class ReaderSettings(  
2.     val readMode: ReaderType.ReadModeType? = null,  
3.     val laserOnTime: Int? = null,  
4.     val inverse1D: ReaderType.Inverse1DType? = null,  
5.     val quietZoneLevelFor1D: ReaderType.QuietZoneLevelType? = null,  
6.     val poorQualityDecodeEffort: ReaderType.PoorQualityDecodeEffortType? = null,  
7. )
```

### 9.1.6 DevSettings

```
1. data class DevSettings(  
2.     val batteryCallback: Boolean? = null  
3. )
```

## 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 래핑하여, 호출만으로 안전한 커맨드를 만들 수 있습니다.

### 10.1 AustralianPostal

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

### 10.2 Aztec

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

### 10.3 Chinese2of5

```
1. object Chinese2of5 {  
2.     @JvmStatic //E4770 미지원  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

### 10.4 Codabar

```
1. object Codabar {  
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 //E4770 미지원  
12.    fun setCLSIEditing(enable: Boolean): SettingCommand  
13.  
14.    @JvmStatic  
15.    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. }

```



```

8.     @JvmStatic
9.     fun setLength1(length: Int): SettingCommand
10.
11.    @JvmStatic
12.    fun setLength2(length: Int): SettingCommand
13.
14.    @JvmStatic //E4770 미지원
15.    fun setReducedQuietZone(enable: Boolean): SettingCommand
16.
17.    @JvmStatic
18.    fun setConvertToCode32(enable: Boolean): SettingCommand
19.
20.    @JvmStatic
21.    fun setFullAscii(enable: Boolean): SettingCommand
22.
23.    @JvmStatic
24.    fun setReportCheckDigit(enable: Boolean): SettingCommand
25.
26.    @JvmStatic
27.    fun setReportCode32Prefix(enable: Boolean): SettingCommand
28.
29.    @JvmStatic //E4770 미지원
30.    fun setSecurityLevel(value: Code39Type.SecurityType): SettingCommand
31.
32.    @JvmStatic
33.    fun setVerifyCheckDigit(enable: Boolean): SettingCommand
34. }

```

### 10.6.1 Code39Type

```

1. object Code39Type {
2.     enum class SecurityType : TypeValue {
3.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
4.     }
5. }

```

## 10.7 Code93

```

1. object Code93 {
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. }

```

## 10.8 Code128

```

1. object Code128 {
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 //E4770 미지원
12.    fun setReducedQuietZone(enable: Boolean): SettingCommand
13.
14.    @JvmStatic //E4770 미지원
15.    fun setIgnoreFnc4(enable: Boolean): SettingCommand
16.
17.    @JvmStatic //E4770 미지원
18.    fun setCheckIsbtTable(enable: Boolean): SettingCommand
19.
20.    @JvmStatic //E4770 미지원
21.    fun setIsbt128ConcatMode(value: Code128Type.ISBT128ConcatModeType): SettingCommand
22.
23.    @JvmStatic //E4770 미지원
24.    fun setSecurityLevel(value: Code128Type.SecurityType): SettingCommand
25.
26.    @JvmStatic //E4770 미지원
27.    fun setEmulationMode(enable: Boolean): SettingCommand
28. }

```

### 10.8.1 Code128Type

```

1. object Code128Type {
2.     enum class SecurityType : TypeValue {
3.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
4.     }
5.
6.     enum class ISBT128ConcatModeType : TypeValue {
7.         ENABLE, DISABLE, AUTO
8.     }
9. }

```

## 10.9 Composite

```

1. object Composite {
2.     @JvmStatic //SE4107, SE5500 미지원
3.     fun setEnable(enable: Boolean): SettingCommand //SE4107, SE5500 미지원
4.
5.     @JvmStatic //E4770 미지원
6.     fun setABEnable(enable: Boolean): SettingCommand
7.
8.     @JvmStatic //E4770 미지원
9.     fun setCEnable(value: Boolean): SettingCommand
10. }

```

```

11.    @JvmStatic //E4770 미지원
12.    fun setTlc39Enable(enable: Boolean): SettingCommand
13.
14.    @JvmStatic //E4770 미지원
15.    fun setUpcCompositeMode(value: CompositeType.UPCCompositeModeType): SettingCommand
16. }

```

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

### 10.9.1 CompositeType

```

1. object CompositeType {
2.     enum class UPCCompositeModeType : TypeValue {
3.         NEVER_LINKED, ALWAYS_LINKED, AUTO_DISCRIMINATE
4.     }
5. }

```

## 10.10 DataMatrix

```

1. object DataMatrix {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setInverse(value: DataMatrixType.InverseType): SettingCommand
7.
8.     @JvmStatic //E4770 미지원
9.     fun setDecodeMirrorImages(value: DataMatrixType.DecodeMirrorImagesType): SettingCommand
10.
11.    @JvmStatic //E4770 미지원
12.    fun setGs1DataMatrix(enable: Boolean): SettingCommand
13. }

```

### 10.10.1 DataMatrixType

```

1. object DataMatrixType {
2.     enum class InverseType : TypeValue {
3.         REGULAR_ONLY, INVERSE_ONLY, AUTO
4.     }
5.
6.     enum class DecodeMirrorImagesType : TypeValue {
7.         NEVER, ALWAYS, AUTO
8.     }
9. }

```

## 10.11 Discrete2Of5

```

1. object Discrete2Of5 {

```

```
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. }
```

## 10.12 DotCode

```
1. object DotCode {
2.     @JvmStatic //E4770 미지원
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic //E4770 미지원
6.     fun setInverse(value: DotCodeType.InverseType): SettingCommand
7.
8.     @JvmStatic //E4770 미지원
9.     fun setMirror(value: DotCodeType.MirrorType): SettingCommand
10.
11.    @JvmStatic //E4770 미지원
12.    fun setPrioritize(enable: Boolean): SettingCommand
13. }
```

### 10.12.1 DotCodeType

```
1. object DotCodeType {
2.     enum class InverseType : TypeValue { ENABLE, DISABLE, AUTODETECT }
3.     enum class MirrorType : TypeValue { REGULAR, INVERSE_ONLY, AUTODETECT }
4. }
```

## 10.13 EAN8

```
1. object EAN8 {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4. }
```

## 10.14 EAN13

```
1. object EAN13 {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4. }
```

## 10.15 Gs1DataBar14

```
1. object Gs1DataBar14 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

## 10.16 Gs1DataBarExpanded

```
1. object Gs1DataBarExpanded {  
2.     @JvmStatic //E4770 미지원  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

## 10.17 Gs1DataBarLimited

```
1. object Gs1DataBarLimited {  
2.     @JvmStatic //E4770 미지원  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic //E4770 미지원  
6.     fun setLimitedSecurityLevel(value: Gs1DataBarLimitedType.SecurityType): SettingCommand  
7. }
```

### 10.17.1 Gs1DataBarLimitedType

```
1. object Gs1DataBarLimitedType {  
2.     enum class SecurityType : TypeValue {  
3.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3  
4.     }  
5. }
```

## 10.18 Gs1128

```
1. object Gs1128 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

## 10.19 HanXin

```

1. object HanXin {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setInverse(value: HanXinType.InverseType): SettingCommand
7. }

```

## 10.20 Interleaved2Of5

```

1. object Interleaved2Of5 {
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 setCheckDigit(value: Interleaved2Of5Type.CheckDigitType): SettingCommand
13.
14.    @JvmStatic
15.    fun setReportCheckDigit(enable: Boolean): SettingCommand
16.
17.    @JvmStatic //E4770 미지원
18.    fun setSecurityLevel(value: Interleaved2Of5Type.SecurityType): SettingCommand
19.
20.    @JvmStatic //E4770 미지원
21.    fun setConvertItf14ToEan13(enable: Boolean): SettingCommand
22.
23.    @JvmStatic //E4770 미지원
24.    fun setReducedQuietZone(enable: Boolean): SettingCommand
25. }

```

### 10.20.1 Interleaved2Of5Type

```

1. object Interleaved2Of5Type {
2.     enum class CheckDigitType : TypeValue {
3.         DISABLE,
4.         ENABLE,
5.         USS_CHECK_DIGIT, //E4770 미지원
6.         OPCC_CHECK_DIGIT //E4770 미지원
7.     }
8.
9.     enum class SecurityType : TypeValue {
10.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
11.     }
12. }

```

## 10.21 ISBT128

```
1. object ISBT128 {  
2.     @JvmStatic //E4770 미지원  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

## 10.22 JapanesePostal

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

## 10.23 Korean3of5

```
1. object Korean3of5 {  
2.     @JvmStatic //E4770 미지원  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

## 10.24 Matrix2of5

```
1. object Matrix2of5 {  
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 //E4770 미지원  
12.    fun setRedundancy(enable: Boolean): SettingCommand  
13.  
14.    @JvmStatic  
15.    fun setReportCheckDigit(enable: Boolean): SettingCommand  
16.  
17.    @JvmStatic  
18.    fun setVerifyCheckDigit(enable: Boolean): SettingCommand  
19. }
```

## 10.25 MaxiCode

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



## 10.26 MicroPdf417

```
1. object MicroPdf417 {  
2.     @JvmStatic //E4770 미지원  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

## 10.27 MicroQrCode

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

## 10.28 MSI

```
1. object MSI {  
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 setCheckDigit(value: MsiType.CheckDigitType): SettingCommand  
13.  
14.    @JvmStatic  
15.    fun setCheckDigitScheme(value: MsiType.CheckDigitSchemeType): SettingCommand  
16.  
17.    @JvmStatic  
18.    fun setReportCheckDigit(enable: Boolean): SettingCommand  
19. }
```

### 10.28.1 MsiType

```
1. object MsiType {  
2.     enum class CheckDigitType : TypeValue {  
3.         ONE_CHECK_DIGIT, TWO_CHECK_DIGITS  
4.     }  
5.  
6.     enum class CheckDigitSchemeType : TypeValue {  
7.         MOD_10_10, MOD_10_11  
8.     }  
9. }
```

## 10.29 NetherlandsKix

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

## 10.30 Pdf417

```
1. object Pdf417 {  
2.     @JvmStatic //E4770 미지원  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

## 10.31 QrCode

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

## 10.32 UKPostal

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

## 10.33 UPCA

```
1. object UPCA {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic  
6.     fun setReportCheckDigit(enable: Boolean): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setPreamble(value: UpcAType.PreambleType): SettingCommand  
10. }
```

```

11.     @JvmStatic
12.     fun setSupplemental2Bit(enable: Boolean): SettingCommand
13.
14.     @JvmStatic
15.     fun setSupplemental5Bit(enable: Boolean): SettingCommand
16. }

```

### 10.33.1 UpcAType

```

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

```

## 10.34 UPCE1

```

1. object UPCE1 {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic //E4770 미지원
6.     fun setConvertToUpcA(enable: Boolean): SettingCommand
7.
8.     @JvmStatic //E4770 미지원
9.     fun setPreamble(value: UpcE1Type.PreambleType): SettingCommand
10.
11.     @JvmStatic //E4770 미지원
12.     fun setReportCheckDigit(enable: Boolean): SettingCommand
13. }

```

### 10.34.1 UpcE1Type

```

1. object UpcE1Type {
2.     enum class PreambleType : TypeValue {
3.         NONE, SYS_CHAR, COUNTRY_SYS_CHAR
4.     }
5. }

```

## 10.35 UPCEan

```

1. object UPCEan {
2.     @JvmStatic
3.     fun setReportEan8CheckDigit(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setReportEan13CheckDigit(enable: Boolean): SettingCommand
7.
8.     @JvmStatic //E4770 미지원
9.     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 setEanSupplemental12(enable: Boolean): SettingCommand
37.
38.     @JvmStatic //E4770 미지원
39.     fun setEanSupplemental15(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 setEan8Supplemental12(enable: Boolean): SettingCommand
49.
50.     @JvmStatic //SE4107, SE5500 미지원
51.     fun setEan13Supplemental12(enable: Boolean): SettingCommand
52.
53.     @JvmStatic //SE4107, SE5500 미지원
54.     fun setEan8Supplemental15(enable: Boolean): SettingCommand
55.
56.     @JvmStatic //SE4107, SE5500 미지원
57.     fun setEan13Supplemental15(enable: Boolean): SettingCommand
58. }

```

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

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

### 10.35.1 UpcEanType

```

1. object UpcEanType {
2.     enum class SupplementalModeType : TypeValue {
3.         IGNORE_SUPPLEMENTALS,
4.         DECODE_WITH_SUPPLEMENTALS,
5.         AUTODISCRIMINATE_SUPPLEMENTALS,
6.         SMART_SUPPLEMENTAL_MODE,
7.         ENABLE_378_379,
8.         ENABLE_978_979,
9.         ENABLE_414_419_434_439,
10.        ENABLE_977,
11.        ENABLE_491,
12.        USER_PROGRAMMABLE_TYPE1,
13.        USER_PROGRAMMABLE_TYPE1_2,
14.        SMART_PLUS_USER_PROGRAMMABLE1,
15.        SMART_PLUS_USER_PROGRAMMABLE1_2
16.    }
17.
18.    enum class SupplementalAimIdFormatType : TypeValue {
19.        SEPARATE, COMBINED, SEPARATE_TRANSMISSION
20.    }
21.
22.    enum class CouponReportType : TypeValue {
23.        OLD_FORMAT, NEW_FORMAT, AUTODISCRIMINATE
24.    }
25.
26.    enum class BooklandType : TypeValue {
27.        ISBN10, ISBN13
28.    }
29. }
```

## 10.36 UPCE

```

1. object UPCE {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setConvertToUpcA(enable: Boolean): SettingCommand
7.
8.     @JvmStatic
9.     fun setPreamble(value: UpcEType.PreambleType): SettingCommand
10.
11.    @JvmStatic
12.    fun setReportCheckDigit(enable: Boolean): SettingCommand
13.
14.    @JvmStatic
15.    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
```

```

6.     fun setAimer(enable: Boolean): SettingCommand
7.
8.     @JvmStatic
9.     fun setIllumination(enable: Boolean): SettingCommand
10.
11.    @JvmStatic
12.    fun setSound(type: SoundType): SettingCommand
13.
14.    @JvmStatic
15.    fun setVibrate(enable: Boolean): SettingCommand
16.
17.    @JvmStatic
18.    fun setLedEnable(enable: Boolean): SettingCommand
19.
20.    @JvmStatic
21.    fun setButtonEnable(enable: Boolean): SettingCommand
22.
23.    @JvmStatic
24.    fun setWithCodeType(enable: Boolean): SettingCommand
25. }

```

### 10.40.1 SoundType

```

1. enum class SoundType : TypeValue {
2.     ON, OFF, MUTE_ON_FAILED_SCAN
3. }

```

## 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

```

1. object ReaderType {
2.     enum class Inverse1DType : TypeValue {
3.         REGULAR_ONLY, INVERSE_ONLY, INVERSE_AUTODETECT
4.     }

```

```
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

```
1. /* M3Utils */
2.
3. @JvmOverloads
4. @JvmStatic
5. fun toUtf8String(bytes: ByteArray, charset: Charset = Charsets.UTF_8): String
```

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

## 2. isDecimalPair

```
1. /* M3Utils */
2.
3. @JvmStatic
4. 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

```
1. /* M3Utils */
2.
3. @JvmStatic
4. 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)

```
1. /* M3Utils */
2.
3. @JvmOverloads
4. @JvmStatic
5. 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)