

# 1. Aztec properties

## 1.1. DEC\_AZTEC\_APPEND\_ENABLED, 0x1a027004

### Description

This property specifies whether Aztec barcodes that contain structured append information are combined into a single result.

When enabled, in order to obtain a result, all pieces of the structured append must be decodable in the same image.

The property value should be set as follows:

0: Disable combination of structured append codes.

1: Enable combination of structured append codes.

Initial Value: 1

### Example

*/\* enable combination of Aztec append codes \*/*

```
DecodeSet( DEC_AZTEC_APPEND_ENABLED, (void *)DEC_CONST_ENABLED );
```

### See also

## 1.2. DEC\_AZTEC\_APPEND\_STRIP\_INFO, 0x1a027005

### Description

This property specifies whether Aztec barcodes that contain structured append information have the information removed before issuing a result.

By enabling this property, the position ID information will be kept where they are encoded in the barcode data. If DEC\_AZTEC\_APPEND\_ENABLED is enabled, this information will be shown at the beginning of each barcode.

The property value should be set as follows:

0: Does not remove append information.

1: Removed append information

Initial Value: 1

### Example

*/\* does not remove information from Aztec append codes \*/*

```
DecodeSet( DEC_AZTEC_APPEND_STRIP_INFO, (void *)DEC_CONST_DISABLED );
```

### See also

## 1.3. DEC\_AZTEC\_ENABLED, 0x1a027001

### Description

This property specifies whether Aztec decoding is enabled during the execution of Decode. Decoding may be separately enabled or disabled for normal and inverse video symbols. A normal video symbol is printed in black on a white substrate. An inverse video symbol is printed in white on a black substrate.

The property value is a bit field defined as follows:

- b0: Enable normal video Aztec decoding
- b1: Enable inverse video Aztec decoding
- b2: Enable Compact Aztec Code decoding
- b3: Enable Full-Size Aztec Code decoding

For example, to decode only Full-Size inverse video Aztec Codes, set the property to 10. The addends for Compact and/or Full-Size decoding are hints to the Decoder, and it is not guaranteed that these will be the only Aztec Code symbols issued.

Initial value: 0

#### **Example**

```
/* enable inverse video Aztec Code decoding only, and give Full-Size hint */  
DecodeSet( DEC_AZTEC_ENABLED, (void *) 0x0a);
```

**See also**

### **1.4. DEC\_AZTEC\_MAX\_LENGTH, 0x1a027003**

#### **Description**

This property controls the maximum length an Aztec Code result must be to be issued.

This property must be greater than or equal to DEC\_AZTEC\_MIN\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

This property does not apply to reader configuration barcodes.

Initial value: 3832

#### **Example**

```
/* set the maximum length Aztec Code to 50 characters */  
DecodeSet( DEC_AZTEC_MAX_LENGTH, (void *) 50);
```

**See also**

### **1.5. DEC\_AZTEC\_MIN\_LENGTH, 0x1a027002**

#### **Description**

This property controls the minimum length an Aztec Code result must be to be issued.

This property must be less than or equal to DEC\_AZTEC\_MAX\_LENGTH so a result can be issued.  
There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

This property does not apply to reader configuration barcodes.

Initial value: 1

**Example**

```
/* set the minimum length Aztec Code to 16 characters */  
DecodeSet( DEC_AZTEC_MIN_LENGTH, (void *) 16);
```

**See also**

## **1.6. DEC\_AZTEC\_SYMBOL\_SIZE, 0x40011202**

**Description**

Controls algorithms that aid in decoding of symbols of different sizes.

Initial value: 1

**See also**

## **1.7. DEC\_ID\_FIX\_EDGE\_2D, 0x40100020**

**Description**

This property is a pass through for the ID property SD\_PROP\_ID\_FIX\_EDGE\_2D. This property ensures that starting from the vertical pass for Aztecs/QR/MaxiCode the duplicate edges of same polarity are consolidated to remove false edges. This feature is useful when the barcode is very noisy and there are several false edges which cause the finder to interpret the signal incorrectly. By eliminating the False Edges, the decoder is more likely to find the finder pattern.

**See also**

SD\_PROP\_ID\_FIX\_EDGE\_2D

## 2. Codablock properties

### 2.1. DEC\_CODABLOCK\_A\_ENABLED, 0x1a030001

#### Description

This property specifies whether Codablock A decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Codablock A decoding

1: Enable Codablock A decoding

When Codablock A and Code 39 decoding are enabled, there is some danger of mistakenly decoding a damaged Codeblock A symbol as a Code 39 symbol. Therefore whenever possible, Code 39 decoding should be disabled when Codablock A decoding is enabled.

Initial Value: 0

#### Example

```
/* enable Codablock A decoding */
```

```
DecodeSet( DEC_CODABLOCK_A_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### 2.2. DEC\_CODABLOCK\_A\_MAX\_LENGTH, 0x1a030003

#### Description

This property specifies the maximum length a Codablock A must be to be issued.

The property must be greater than or equal to DEC\_CODABLOCK\_A\_MIN\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 2048

#### Example

```
/* set maximum length Codablock A to 200 characters */
```

```
DecodeSet( DEC_CODABLOCK_A_MAX_LENGTH, (void *) 200);
```

**See also**

### 2.3. DEC\_CODABLOCK\_A\_MIN\_LENGTH, 0x1a030002

#### Description

This property specifies the minimum length a Codablock A must be to be issued.

The property must be less than or equal to DEC\_CODABLOCK\_A\_MAX\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 1

**Example**

```
/* set minimum length Codablock A to 10 characters */  
DecodeSet( DEC_CODABLOCK_A_MIN_LENGTH, (void *) 10);
```

**See also**

## 2.4. DEC\_CODABLOCK\_F\_ENABLED, 0x1a023001

**Description**

This property specifies whether Codablock F decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Codablock F decoding

1: Enable Codablock F decoding

When Codablock F and Code 128 decoding are enabled, there is some danger of mistakenly decoding a damaged Codablock F symbol as a Code 128 symbol. Therefore whenever possible, Code 128 decoding should be disabled when Codablock F decoding is enabled.

Initial Value: 0

**Example**

```
/* enable Codablock F decoding */  
DecodeSet( DEC_CODABLOCK_F_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

## 2.5. DEC\_CODABLOCK\_F\_MAX\_LENGTH, 0x1a023003

**Description**

This property specifies the maximum length a Codablock F must be to be issued.

The property must be greater than or equal to DEC\_CODABLOCK\_F\_MIN\_LENGTH so a result can be issued. There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 60

**Example**

```
/* set maximum length Codablock F to 200 characters */
```

```
DecodeSet( DEC_CODABLOCK_F_MAX_LENGTH, (void *) 200);
```

**See also**

## **2.6. DEC\_CODABLOCK\_F\_MIN\_LENGTH, 0x1a023002**

### **Description**

This property specifies the minimum length a Codablock F must be to be issued.

The property must be less than or equal to DEC\_CODABLOCK\_F\_MAX\_LENGTH so a result can be issued.  
There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 1

### **Example**

```
/* set minimum length Codablock F to 10 characters */
```

```
DecodeSet( DEC_CODABLOCK_F_MIN_LENGTH, (void *) 10);
```

**See also**

### 3. Code 11 properties

#### 3.1. DEC\_CODE11\_CHECK\_DIGIT\_MODE, 0x40011802

##### Description

This property specifies how the Code 11 check digit is to be handled during the execution of Decode().

The property value should be set as follows:

- 0: Two check digits verified.
- 1: One check digit verified.
- 2: Two check digits verified and stripped from result data.
- 3: One check digit verified and stripped from result data.

Initial Value: 0

##### Example

```
/* enabled verification of single Code 11 check digit and strip from result */
DecodeSet( DEC_CODE11_CHECK_DIGIT_MODE, (void *)3);
```

See also

#### 3.2. DEC\_CODE11\_ENABLED, 0x1a01e001

##### Description

This property specifies whether Code 11 decoding is enabled during the execution of Decode.

The property value should be set as follows:

- 0: Disable Code 11 decoding
- 1: Enable Code 11 decoding

Initial Value: 0

##### Example

```
/* enable Code 11 decoding */
DecodeSet( DEC_CODE11_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

#### 3.3. DEC\_CODE11\_IMPROVE\_BOUNDS, 0x40011803

##### Description

This property specifies whether additional processing should be performed in an attempt to improve the bounds of a Code 11 symbol before a result is issued.

In order to improve the bounds of a Code 11 symbol, the amount of time before the symbol may be significantly

increased.

The property value should be set as follows:

0: Disable improved bounds processing.

1: Enable improved bounds processing.

Initial Value: 0

#### **Example**

```
/* enable Code 11 improve bounds */
```

```
DecodeSet( DEC_CODE11_IMPROVE_BOUNDS, (void *)DEC_CONST_ENABLED );
```

#### **See also**

### **3.4. DEC\_CODE11\_MAX\_LENGTH, 0x1a01e003**

#### **Description**

This property specifies the maximum length a Code 11 must be to be issued.

The property must be greater than or equal to DEC\_CODE11\_MIN\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 80

#### **Example**

```
/* set maximum length Code 11 to 20 characters */
```

```
DecodeSet( DEC_CODE11_MAX_LENGTH, (void *) 20);
```

#### **See also**

### **3.5. DEC\_CODE11\_MIN\_LENGTH, 0x1a01e002**

#### **Description**

This property specifies the minimum length a Code 11 must be to be issued.

The property must be less than or equal to DEC\_CODE11\_MAX\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 4

#### **Example**

```
/* set minimum length Code 11 to 10 characters */
```

```
DecodeSet( DEC_CODE11_MIN_LENGTH, (void *) 10);
```

#### **See also**





## 4. Code 128 properties

### 4.1. DEC\_C128\_FNC1\_SUBSTITUTE, 0x1a014006

**Description**

This property specifies the character to be used as a replacement of the FNC1 character when not in first or second position.

This applies to code 128 and GS1-128 labels.

Default: 0x1D

**Example**

```
/* set '#' character as FNC1 substitute */
```

```
DecodeSet( DEC_C128_FNC1_SUBSTITUTE, (void *) 0x23 );
```

**See also**

### 4.2. DEC\_C128\_FNC\_TRANSMIT, 0x1a014007

**Description**

This property specifies whether a "function" codeword, if present, is transmitted in the data for a Code 128 symbol.

"Function" codewords are replaced with an equivalent 0x8n value. For example, a FNC1 is replaced with 0x81.

Note that not all Code 128 symbols contain function characters.

This property also applies to GS1-128 symbols, as they are defined by having a FNC1 codeword at the beginning of the symbol.

The property value should be set as follows:

0: Do not transmit "function" characters.

1: Transmit "function" characters.

Initial Value: 0

**See also**

### 4.3. DEC\_C128\_IMPROVE\_BOUNDS, 0x40010208

**Description**

This property specifies whether additional processing should be performed in an attempt to improve the bounds of a Code 128 symbol before a result is issued.

In order to improve the bounds of a Code 128 symbol, the amount of time before the symbol may be significantly increased.

The property value should be set as follows:

0: Disable improved bounds processing.

1: Enable improved bounds processing.

Initial Value: 0

#### Example

```
/* enable Code 128 improve bounds */
```

```
DecodeSet( DEC_CODE128_IMPROVE_BOUNDS, (void *)DEC_CONST_ENABLED );
```

#### See also

### 4.4. DEC\_C128\_ISBT\_ALIGNED, 0x1a01400b

#### Description

This property specifies the codes alignment requirement for the ISBT concatenation.

The property value should be set as follows:

0: can concatenate ISBT codes from different image, whatever the relative position of the 2 codes

1: can concatenate only codes that are horizontally aligned, so can concatenate codes that are decoded in the same image

Initial Value: 0

#### Example

```
/* Disable the ISBT alignment requirement for ISBT concatenation */
```

```
DecodeSet( DEC_C128_ISBT_ALIGNED, (void *)0x00);
```

#### See also

DEC\_C128\_ISBT\_ENABLED

DEC\_C128\_ISBT\_CONCATENATION

DEC\_C128\_ISBT\_STRICT

DEC\_C128\_USE\_ALT\_ISBT\_ID

### 4.5. DEC\_C128\_ISBT\_CONCATENATION, 0x1a014009

#### Description

This property specifies the ISBT concatenation mode according to ISBT specification recommendations.

The property value should be set as follows:

0: None (concatenated read prohibited)

1: Required (concatenated read required)

2: Optional (concatenated read permitted but not required)

Initial Value: 1

### Example

```
/* Set ISBT concatenation to 'required' */  
DecodeSet( DEC_C128_ISBT_CONCATENATION, (void *)0x01);
```

### See also

DEC\_C128\_ISBT\_ENABLED  
DEC\_C128\_ISBT\_STRICT  
DEC\_C128\_ISBT\_ALIGNED  
DEC\_C128\_USE\_ALT\_ISBT\_ID

## 4.6. DEC\_C128\_ISBT\_ENABLED, 0x1a014005

### Description

This property specifies whether ISBT reading is enabled.

Note : when disabled, ISBT codes can be read if the code 128 is enabled. When enabled, ISBT codes are read whatever the code 128 is enabled or not.

For full explanation of the requirements for ISBT concatenation, see the United States Industry Consensus Standard for the Uniform Labeling of Blood and Blood Components using ISBT128 document, or visit <http://iccbba.com>.

The property value should be set as follows:

- 0: Disable ISBT reading
- 1: Enable ISBT reading

Initial Value: 0

### Example

```
/* Enabled Code 128 ISBT */  
DecodeSet( DEC_C128_ISBT_ENABLED, (void *)0x01);
```

### See also

DEC\_C128\_ISBT\_CONCATENATION  
DEC\_C128\_ISBT\_STRICT  
DEC\_C128\_ISBT\_ALIGNED  
DEC\_C128\_USE\_ALT\_ISBT\_ID

## 4.7. DEC\_C128\_ISBT\_STRICT, 0x1a01400a

### Description

This property specifies the rules to apply for ISBT concatenation.

The property value should be set as follows:

- 0: can concatenate any ISBT codes
- 1: can concatenate only ISBT codes specified by specification

Initial value 1

#### Example

```
/* Enabled strict ISBT concatenation */
DecodeSet( DEC_C128_ISBT_STRICT, (void *)0x01);
```

#### See also

DEC\_C128\_ISBT\_ENABLED  
 DEC\_C128\_ISBT\_CONCATENATION  
 DEC\_C128\_ISBT\_ALIGNED  
 DEC\_C128\_USE\_ALT\_ISBT\_ID

## 4.8. DEC\_C128\_OUT\_OF\_SPEC\_SYMBOL, 0x40010203

### Description

This property enables enhancements for reading difficult Code 128 bar codes during the execution of the Decode.

The property value is a bit field defined as follows:

b0: Enable Code 128 Enhancement for reading codes whose bars have inconsistent width from top to bottom.

NOTE: This setting only works when DEC\_USE\_MLD for Code 128 is disabled. Also, this setting can affect read performance when DEC\_GENERAL\_IMPROVEMENTS bit +2 is set.

b1: Enable Code 128 Enhancement for reading codes with extreme bar growth (i.e. over inking)

b2: Enable reading of Code 128 barcodes with Out of Spec Start patterns (1st bar is 1x). NOTE: b0 and b2 cannot be activated at the same time (b0 won't work as expected).

b3: Increase the codeword to codeword length tolerance. NOTE: Using this setting can decrease reading performances on low contrast or damaged barcodes.

b4: Enable reading poor quality Code 128s whose bar and space widths fall outside standard tolerances. Setting this bit may increase the likelihood of misreads.

Initial Value: 0

#### See also

DEC\_USE\_MLD  
 DEC\_GENERAL\_IMPROVEMENTS

## 4.9. DEC\_C128\_PARTIAL, 0x40010207

### Description

This property specifies whether a Code 128 symbol should be issued when only part of the symbol is present.

This property should only be used if it is absolutely necessary to receive decode results when the full symbol is not present. This property specifies the minimum number of characters that must be present to issue a partial result.

The minimum characters needed before a partial result is issued is 4 characters. The ModifierEx value will specify whether the issued result is a partial result.

Note: When issuing partial symbols it is possible that the entire symbol may be issued after a partial result of the same barcode. Enabling this property may adversely affect read rates on marginal symbols.

Initial Value:0

#### Example

```
/* issue partial Code 128 symbols with at least 10 characters */  
DecodeSet( DEC_CODE128_PARTIAL, (void *)10);
```

See also

### 4.10. DEC\_C128\_SHORT\_MARGIN, 0x40010202

#### Description

This property specifies whether substandard length margins (i.e. quiet zones) should be allowed for Code 128 symbols during the execution of Decode().

When enabled, a substandard length margin is allowed on either end of a Code 128 symbol, but not both. This property must be explicitly set to allow short margins on both ends. Enabling this property is not recommended unless absolutely necessary.

The property value should be set as follows:

- 0: Disable short margin decoding.
- 1: Enable short margin decoding.
- 2: Enable short margin decoding on both ends.

Initial Value: 1

#### Example

```
/* enable short margin Code 128 symbols */  
DecodeSet( DEC_CODE128_SHORT_MARGIN, (void *)DEC_CONST_ENABLED);
```

See also

### 4.11. DEC\_C128\_SUPPRESS\_CODABLOCK\_CONFLICT, 0x40010206

#### Description

This property specifies whether a Code 128 symbol should be issued if Codablock F is enabled and the symbol appears to be part of a Codablock F symbol.

This property should only be used when both DEC\_CODE128\_ENABLED and DEC\_CODABLOCK\_A\_ENABLED are both enabled. A Code 128 symbol in which the first codeword is a Start A character, the second codeword is a SHIFT, CODE B, or CODE C character, and the third codeword is a valid Codablock F row indicator.

The property value should be set as follows:

- 0: Issue symbols that may be part of a Codablock F.

1: Do not issue symbols that may be part of a Codablock F.

Initial Value: 1

#### Example

```
/* issue Code 128 symbols that may be part of a Codablock F */  
DecodeSet( DEC_CODE128_SUPPRESS_CODABLOCK_CONFLICT, (void *)DEC_CONST_DISABLED);
```

#### See also

### 4.12. DEC\_C128\_USE\_ALT\_ISBT\_ID, 0x1A01400C

#### Description

This property specifies whether ISBT codes should be discriminated from code 128. When enabled, ISBT codes will have a specific code ID, different from code 128, and cannot be read as regular code 128 when DEC\_C128\_ISBT\_ENABLED property is disabled.

The property value should be set as follows:

0: disabled

1: enabled

Initial Value: 0

#### Example

```
/* Disable alternate ID for ISBT codes */  
DecodeSet( DEC_C128_USE_ALT_ISBT_ID, (void *)0x00);
```

#### See also

DEC\_C128\_ISBT\_ENABLED

DEC\_C128\_ISBT\_CONCATENATION

DEC\_C128\_ISBT\_ALIGNED

DEC\_C128\_ISBT\_STRICT

### 4.13. DEC\_CODE128\_APPEND\_ENABLED, 0x1a014004

#### Description

This property specifies whether Code 128 codes with append information are combined into a single result.

When enabled, each symbol with Code 128 append information must be decoded 1 at a time and will be concatenated to any previous data and the CB\_ResultNotify() callback will be called.

Once a Code 128 without append information is decoded, all data scanned will be combined into a single result, and that result will be issued through the CB\_Result() callback.

If a different symbology is decoded, all data previous data will be cleared.

Initial Value: 0

#### **Example**

```
/* enable the combination of Code 128 code with append information */  
DecodeSet( DEC_CODE128_APPEND_ENABLED, (void *)DEC_CONST_ENABLED);
```

#### **See also**

### **4.14. DEC\_CODE128\_ENABLED, 0x1a014001**

#### **Description**

This property specifies whether Code 128 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Code 128 decoding.

1: Enable Code 128 decoding.

Initial Value: 0

#### **Example**

```
/* enable Code 128 decoding */  
DecodeSet( DEC_CODE128_ENABLED, (void *) 1);
```

#### **See also**

### **4.15. DEC\_CODE128\_MAX\_LENGTH, 0x1a014003**

#### **Description**

This property specifies the maximum length a Code 128 must be to be issued.

The property must be greater than or equal to DEC\_CODE128\_MIN\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 80

#### **Example**

```
/* set maximum length Code 128 to 20 characters */  
DecodeSet( DEC_CODE128_MAX_LENGTH, (void *) 20);
```

#### **See also**

### **4.16. DEC\_CODE128\_MIN\_LENGTH, 0x1a014002**

#### **Description**



This property specifies the minimum length a Code 128 must be to be issued.

The property must be less than or equal to DEC\_CODE128\_MAX\_LENGTH so a result can be issued.  
There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 1

#### Example

```
/* set minimum length Code 128 to 10 characters */  
DecodeSet( DEC_CODE128_MIN_LENGTH, (void *) 10);
```

See also

### 4.17. DEC\_CODE128\_PARITY\_THRESHOLD, 0x0A01400D

#### Description

Modified Code 128 decoder to use the DEC\_CODE128\_PARITY\_THRESHOLD DCL property to improve decode security. can be used to improve Code 128 decode security at the expense of aggressiveness. The property defaults to 255 (disabled). Lower values improve security, especially on codes with inconsistent bar and space widths.

#### Example

```
RSI_DeclareSet( DEC_CODE128_PARITY_THRESHOLD, void *data )
```

See also

### 4.18. DEC\_GS1\_128\_ENABLED, 0x1a015001

#### Description

This property specifies whether GS1 128 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable GS1 128 decoding

1: Enable GS1 128 decoding

Initial Value: 0

#### Example

```
/* enable GS1 128 decoding */  
DecodeSet( DEC_GS1_128_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

#### **4.19. DEC\_GS1\_128\_MAX\_LENGTH, 0x1a015003**

**Description**

Maximum data length to issue a result for GS1-128.

Default: 0

**See also**

DEC\_GS1\_128\_MIN\_LENGTH

#### **4.20. DEC\_GS1\_128\_MIN\_LENGTH, 0x1a015002**

**Description**

Minimum data length to issue a result for GS1-128.

Default: 1

**See also**

DEC\_GS1\_128\_MAX\_LENGTH

## 5. Codabar properties

### 5.1. DEC\_CB\_HIGH\_DENSITY, 0x40010106

#### Description

Enable High Density 1PPM codabar search.

#### See also

### 5.2. DEC\_CODABAR\_CHECK\_DIGIT\_MODE, 0x1a01f005

#### Description

This property specifies how the Codabar check digit is to be handled during the execution of Decode().

The property value should be set as follows:

- 0: Disable checksum verification.
- 1: Enable checksum verification.
- 2: Enable checksum verification and strip digit from result.
- 3: Enable 7DR checksum verification.
- 4: Enable 7DR checksum verification and strip digit from result.

Initial Value: 0

#### Example

```
/* enabled verification of Codabar check digit and strip from result */
DecodeSet( DEC_CODABAR_CHECK_DIGIT_MODE, (void *)DEC_CONST_STRIPPED);
```

#### See also

### 5.3. DEC\_CODABAR\_CONCAT\_ENABLED, 0x1a01f007

#### Description

This property specifies whether Codabar codes with concatenation information are combined into a single result.

When enabled, in order to obtain a result, all pieces of the structured append must be decodable in the same image.

- 0: Codabar Concatenation Off
- 1: Codabar Concatenation On
- 2: Codabar Concatenation Required

Initial Value: 0

#### Example

```
/* enable the combination of Codabar code with concatenation information */
DecodeSet( DEC_CODABAR_CONCAT_ENABLED, (void *)0x01);
```

**See also**

#### **5.4. DEC\_CODABAR\_ENABLED, 0x1a01f001**

##### **Description**

This property specifies whether Codabar decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Codabar decoding

1: Enable Codabar decoding

Initial Value: 0

##### **Example**

```
/* enable Codabar decoding */
```

```
DecodeSet( DEC_CODABAR_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

#### **5.5. DEC\_CODABAR\_IMPROVE\_BOUNDS, 0x40010104**

##### **Description**

This property specifies whether additional processing should be performed in an attempt to improve the bounds of a Codabar symbol before a result is issued.

In order to improve the bounds of a Codabar symbol, the amount of time before the symbol may be significantly increased.

The property value should be set as follows:

0: Disable improved bounds processing.

1: Enable improved bounds processing.

Initial Value: 0

##### **Example**

```
/* enable Codabar improve bounds */
```

```
DecodeSet( DEC_CODABAR_IMPROVE_BOUNDS, (void *)DEC_CONST_ENABLED );
```

**See also**

#### **5.6. DEC\_CODABAR\_MAX\_LENGTH, 0x1a01f003**

### **Description**

This property specifies the maximum length a Codabar must be to be issued.

The property must be greater than or equal to DEC\_CODABAR\_MIN\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 60

### **Example**

```
/* set maximum length Codabar to 20 characters */  
DecodeSet( DEC_CODABAR_MAX_LENGTH, (void *) 20);
```

### **See also**

## **5.7. DEC\_CODABAR\_MIN\_LENGTH, 0x1a01f002**

### **Description**

This property specifies the minimum length a Codabar must be to be issued.

The property must be less than or equal to DEC\_CODABAR\_MAX\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 4

### **Example**

```
/* set minimum length Codabar to 10 characters */  
DecodeSet( DEC_CODABAR_MIN_LENGTH, (void *) 10);
```

### **See also**

## **5.8. DEC\_CODABAR\_SHORT\_MARGIN, 0x40010103**

### **Description**

This property specifies whether substandard length margins (i.e. quiet zones) should be allowed for Codabar symbols during the execution of Decode().

When enabled, a substandard length margin is allowed on either end of a Codabar symbol, but not both.

Enabling this property is not recommended unless absolutely necessary.

The property value should be set as follows:

0: Disable short margin decoding.

1: Enable short margin decoding.

Initial Value: 1

**Example**

```
/* enable short margin Codabar symbols */
```

```
DecodeSet( DEC_CODABAR_SHORT_MARGIN, (void *)DEC_CONST_ENABLED);
```

**See also**

## 5.9. DEC\_CODABAR\_START\_STOP\_TRANSMIT, 0x1a01f004

**Description**

This property specifies whether the start and stop characters are included with the result data.

With DEC\_CODABAR\_CONCAT\_ENABLED enabled, only the start character from the first code and the stop character from the last code will be included in the data.

The property value should be set as follows:

0: Start and stop characters are not included.

1: Start and stop characters are included.

Initial Value: 0

**Example**

```
/* enabled transmission of Codabar start and stop characters */
```

```
DecodeSet( DEC_CODABAR_START_STOP_TRANSMIT, (void *)DEC_CONST_ENABLED);
```

**See also**

## 5.10. DEC\_CODABAR\_VESTA, 0xfd000007

**Description**

This property specifies whether Vesta decoding for Codabar is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Vesta decoding for Codabar.

1: Enable Vesta decoding for Codabar.

Initial Value: 0

**See also**

## 6. Code 39 properties

### 6.1. DEC\_CODE39\_APPEND\_ENABLED, 0x1a016005

**Description**

This property specifies whether Code 39 codes with append information are combined into a single result.

When enabled, each symbol with Code 39 append information must be decoded 1 at a time and will be concatenated to any previous data and the CB\_ResultNotify() callback will be called.

Once a Code 39 without append information is decoded, all data scanned will be combined into a single result, and that result will be issued through the CB\_Result() callback.

If a different symbology is decoded, all data previous data will be cleared.

Initial Value: 0

**Example**

```
/* enable the combination of Code 39 code with append information */
```

```
DecodeSet( DEC_CODE39_APPEND_ENABLED, (void *)DEC_CONST_ENABLED);
```

**See also**

### 6.2. DEC\_CODE39\_BASE32\_ENABLED, 0x1a016008

**Description**

This property specifies whether base 32 interpretation should be done before the Code 39 result is issued.

A result will be issued regardless of this setting. When enabled, this property will allow 6 characters results to be translated to their base 32 equivalents.

The property value should be set as follows:

0: Disable.

1: Enable

Initial Value: 0

**Example**

```
/* enable base 32 interpretation for Code 39 symbols */
```

```
DecodeSet( DEC_CODE39_BASE32_ENABLED, (void *)1);
```

**See also**

DEC\_CODE39\_ENABLED

### 6.3. DEC\_CODE39\_CHECK\_DIGIT\_MODE, 0x1a016004

**Description**

This property specifies how the Code 39 check digit is to be handled during the execution of Decode().

The property value should be set as follows:

- 0: Disable checksum verification.
- 1: Enable checksum verification.
- 2: Enable checksum verification and strip digit from result.

Initial Value: 0

#### Example

```
/* enabled verification of Code 39 check digit and strip from result */  
DecodeSet( DEC_CODE39_CHECK_DIGIT_MODE, (void *)DEC_CONST_STRIPPED);
```

**See also**

### 6.4. DEC\_CODE39\_ENABLED, 0x1a016001

#### Description

This property specifies whether Code 39 decoding is enabled during the execution of Decode.

The property value should be set as follows:

- 0: Disable Code 39 decoding
- 1: Enable Code 39 decoding

Initial Value: 0

#### Example

```
/* enable Code 39 decoding */  
DecodeSet( DEC_CODE39_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### 6.5. DEC\_CODE39\_FULL\_ASCII\_ENABLED, 0x1a016006

#### Description

This property specifies whether Code 39 full ASCII decoding is enabled during the execution of Decode().

The property value should be set as follows:

- 0: Disable full ASCII decoding.
- 1: Enable full ASCII decoding.

Initial Value: 0

#### Example

```
/* enable full ASCII Code 39 decoding */  
DecodeSet( DEC_CODE39_FULL_ASCII_ENABLED, (void *)DEC_CONST_ENABLED);
```



**See also**

## 6.6. DEC\_CODE39\_IMPROVE\_BOUNDS, 0x40010310

### Description

This property specifies whether additional processing should be performed in an attempt to improve the bounds of a Code 39 symbol before a result is issued.

In order to improve the bounds of a Code 39 symbol, the amount of time before the symbol may be significantly increased.

The property value should be set as follows:

0: Disable improved bounds processing.

1: Enable improved bounds processing.

Initial Value: 0

### Example

```
/* enable Code 39 improve bounds */
```

```
DecodeSet( DEC_CODE39_IMPROVE_BOUNDS, (void *)DEC_CONST_ENABLED );
```

**See also**

## 6.7. DEC\_CODE39\_MAX\_LENGTH, 0x1a016003

### Description

This property specifies the maximum length a Code 39 must be to be issued.

The property must be greater than or equal to DEC\_CODE39\_MIN\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 48

### Example

```
/* set maximum length Code 39 to 20 characters */
```

```
DecodeSet( DEC_CODE39_MAX_LENGTH, (void *) 20);
```

**See also**

## 6.8. DEC\_CODE39\_MIN\_LENGTH, 0x1a016002

### Description

This property specifies the minimum length a Code 39 must be to be issued.

The property must be less than or equal to DEC\_CODE39\_MAX\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 1

#### Example

```
/* set minimum length Code 39 to 10 characters */
DecodeSet( DEC_CODE39_MIN_LENGTH, (void *) 10);
```

**See also**

## 6.9. DEC\_CODE39\_PARTIAL, 0x40010309

### Description

This property specifies whether a Code 39 symbol should be issued when only part of the symbol is present.

This property should only be used if it is absolutely necessary to receive decode results when the full symbol is not present.

This property specifies the minimum number of characters that must be present to issue a partial result. The minimum characters needed before a partial result is issued is 4 characters.

The ModifierEx value will specify whether the issued result is a partial result.

Note: When issuing partial symbols it is possible that the entire symbol may be issued after a partial result of the same barcode.

Enabling this property may adversely affect read rates on marginal symbols.

When DEC\_CODE39\_FULL\_ASCII\_ENABLED is disabled, partial results may be issued after either a start or stop character.

When DEC\_CODE39\_FULL\_ASCII\_ENABLED is enabled, partial results will only be issued from the start character.

Initial Value: 0

### Example

```
/* issue partial Code 39 symbols with at least 10 characters */
DecodeSet( DEC_CODE39_PARTIAL, (void *)10);
```

**See also**

## 6.10. DEC\_CODE39\_SHORT\_MARGIN, 0x40010304

### Description

This property specifies whether substandard length margins (i.e. quiet zones) should be allowed for Code 39

symbols during the execution of Decode().

When enabled, a substandard length margin is allowed on either end of a Code 39 symbol, but not both. This property must be explicitly set to allow short margins on both ends. Enabling this property is not recommended unless absolutely necessary.

The property value should be set as follows:

- 0: Disable short margin decoding.
- 1: Enable short margin decoding.
- 2: Enable short margin decoding on both ends.

Initial Value: 1

#### Example

```
/* enable short margin Code 39 symbols */
DecodeSet( DEC_CODE39_SHORT_MARGIN, (void *)DEC_CONST_ENABLED);
```

**See also**

## 6.11. DEC\_CODE39\_START\_STOP\_TRANSMIT, 0x1a016007

### Description

This property specifies whether Code 39 start and stop characters are included with the result data.

Initial Value: 0

### Example

```
/* enabled transmission of Code 39 start and stop characters */
DecodeSet( DEC_CODE39_START_STOP_TRANSMIT, (void *)DEC_CONST_ENABLED);
```

**See also**

## 6.12. DEC\_CODE39\_SUPPRESS\_CODABLOCK\_CONFLICT, 0x40010306

### Description

This property specifies whether a Code 39 symbol should be issued if Codablock A is enabled and the symbol appears to be part of a Codablock A symbol.

This property should only be used when both DEC\_CODE39\_ENABLED and DEC\_CODABLOCK\_A\_ENABLED are both enabled.

A Code 39 symbol in which the first and last codewords are the same may be part of a Codablock A, and will be suppressed with this property enabled.

This is not a terribly unique situation, so users are cautioned that perfectly valid Code 39 symbols may be suppressed.

The property value should be set as follows:

0: Issue symbols that may be part of a Codablock A.

1: Do not issue symbols that may be part of a Codablock A.

Initial Value: 1

**Example**

*/\* do not issue Code 39 symbols that may be part of a Codablock A \*/*

*DecodeSet( DEC\_CODE39\_SUPPRESS\_CODABLOCK\_CONFLICT, (void \*)DEC\_CONST\_ENABLED);*

**See also**

### 6.13. DEC\_CODE39\_UNCONV\_INTER\_CHAR, 0x40010313

**Description**

This property specifies whether Code 39 barcodes with unconventional intercharacter gaps can be read.

The property value should be set as follows:

0: Disable reading of Code 39 barcodes with unconventional intercharacter gaps.

1: Enable reading of Code 39 barcodes with unconventional intercharacter gaps.

Initial Value: 0

**Example**

*/\* Read Code 39 symbols with unconventional intercharacter gaps \*/*

*DecodeSet( DEC\_CODE39\_UNCONV\_INTER\_CHAR, (void \*)1);*

**See also**

DEC\_FLD\_CODE39\_UNCONV\_INTER\_CHAR

### 6.14. DEC\_FLD\_CODE39\_UNCONV\_INTER\_CHAR, 0xfd004009

**Description**

This property specifies whether Code 39 barcodes with unconventional intercharacter gaps can be read.

The property value should be set as follows:

0: Disable reading of Code 39 barcodes with unconventional intercharacter gaps.

1: Enable reading of Code 39 barcodes with unconventional intercharacter gaps.

2: Enable reading of Code 39 barcodes with small or missing intercharacter gaps.

Initial Value: 0

**Example**

*/\* Read Code 39 symbols with unconventional intercharacter gaps \*/*

*DecodeSet( DEC\_FLD\_CODE39\_UNCONV\_INTER\_CHAR, (void \*)1);*

**See also**

DEC\_CODE39\_UNCONV\_INTER\_CHAR

## 7. Code 93 properties

### 7.1. DEC\_CODE93\_APPEND\_ENABLED, 0x1a01d004

#### Description

This property specifies whether Code 93 codes with append information are combined into a single result.

When enabled, each symbol with Code 93 append information must be decoded 1 at a time and will be concatenated to any previous data and the CB\_ResultNotify() callback will be called.

Once a Code 93 without append information is decoded, all data scanned will be combined into a single result, and that result will be issued through the CB\_Result() callback.

If a different symbology is decoded, all data previous data will be cleared.

Initial Value: 0

#### Example

*/\* enable the combination of Code 93 code with append information \*/*

```
DecodeSet( DEC_CODE93_APPEND_ENABLED, (void *)DEC_CONST_ENABLED);
```

#### See also

### 7.2. DEC\_CODE93\_ENABLED, 0x1a01d001

#### Description

This property specifies whether Code 93 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Code 93 decoding.

1: Enable Code 93 decoding.

Initial Value: 0

#### Example

*/\* enable Code 93 decoding \*/*

```
DecodeSet( DEC_CODE93_ENABLED, (void *) DEC_CONST_ENABLED);
```

#### See also

### 7.3. DEC\_CODE93\_HIGH\_DENSITY, 0x40011104

#### Description

This property improves decoding of high density Code 93 barcodes. Enabling this property increases decoding time.

The property value should be set as follows:

0: Disable Code 93 high density decoding improvements.

1: Enable Code 93 high density decoding improvements.

Initial Value: 0

#### Example

```
/* Enable Code 93 high density decoding improvements */  
DecodeSet( DEC_CODE93_HIGH_DENSITY, (void *) DEC_CONST_ENABLED);
```

#### See also

DEC\_CODE93\_ENABLED

## 7.4. DEC\_CODE93\_IMPROVE\_BOUNDS, 0x40011103

### Description

This property specifies whether additional processing should be performed in an attempt to improve the bounds of a Code 93 symbol before a result is issued.

In order to improve the bounds of a Code 93 symbol, the amount of time before the symbol may be significantly increased.

The property value should be set as follows:

0: Disable improved bounds processing.

1: Enable improved bounds processing.

Initial Value: 0

### Example

```
/* enable Code 93 improve bounds */  
DecodeSet( DEC_CODE93_IMPROVE_BOUNDS, (void *)DEC_CONST_ENABLED );
```

### See also

## 7.5. DEC\_CODE93\_MAX\_LENGTH, 0x1a01d003

### Description

This property specifies the maximum length a Code 93 must be to be issued.

The property must be greater than or equal to DEC\_CODE93\_MIN\_LENGTH so a result can be issued.

There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 80

### Example

```
/* set maximum length Code 93 to 20 characters */  
DecodeSet( DEC_CODE93_MAX_LENGTH, (void *) 20);
```

**See also**

## 7.6. DEC\_CODE93\_MIN\_LENGTH, 0x1a01d002

### Description

This property specifies the minimum length a Code 93 must be to be issued.

The property must be less than or equal to DEC\_CODE93\_MAX\_LENGTH so a result can be issued.  
There is no internal check to ensure this requirement to allow asynchronous configuration of this property.

Initial Value: 0

### Example

```
/* set minimum length Code 93 to 10 characters */
DecodeSet( DEC_CODE93_MIN_LENGTH, (void *) 10);
```

**See also**

## 7.7. DEC\_CODE93\_SHORT\_MARGIN, 0x40011102

### Description

This property specifies whether substandard length margins (i.e. quiet zones) should be allowed for Code 93 symbols during the execution of Decode().

When enabled, a substandard length margin is allowed on either end of a Code 93 symbol, but not both.  
Enabling this property is not recommended unless absolutely necessary.

The property value should be set as follows:

0: Disable short margin decoding.

1: Enable short margin decoding.

Initial Value: 1

### Example

```
/* enable short margin Code 93 symbols */
DecodeSet( DEC_CODE93_SHORT_MARGIN, (void *)DEC_CONST_ENABLED);
```

**See also**



## 8. TLC 39 properties

### 8.1. DEC\_TLC39\_ENABLED, 0x1a017001

#### Description

This property specifies whether TLC39 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable TLC39 decoding

1: Enable TLC39 decoding

Initial Value: 0

#### Example

```
/* enable TLC39 decoding */
```

```
DecodeSet( DEC_TLC39_ENABLED, (void *) DEC_CONST_ENABLED);
```

#### See also

## 9. Hong Kong 2 of 5 properties

### 9.1. DEC\_HK25\_ENABLED, 0x1a02c001

#### Description

This property specifies whether Hong Kong 2 of 5 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Hong Kong 2 of 5 decoding

1: Enable Hong Kong 2 of 5 decoding

Initial Value: 0

#### Example

```
/* enable Hong Kong 2 of 5 decoding */
```

```
DecodeSet( DEC_HK25_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

### 9.2. DEC\_HK25\_IMPROVE\_BOUNDS, 0x40012603

#### Description

Controls additional processing to improve bounds.

See also

### 9.3. DEC\_HK25\_MAX\_LENGTH, 0x1a02c003

#### Description

Maximum data length to issue a result for Hong Kong 2 of 5.

See also

### 9.4. DEC\_HK25\_MIN\_LENGTH, 0x1a02c002

#### Description

Minimum data length to issue a result for Hong Kong 2 of 5.

See also

## 10. Hanxin properties

### 10.1. DEC\_HANXIN\_ENABLED, 0x1a02b001

#### Description

This property specifies whether HanXin decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable HanXin decoding

1: Enable HanXin decoding

Initial Value: 0

#### Example

```
/* enable HanXin decoding */
```

```
DecodeSet( DEC_HANXIN_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### 10.2. DEC\_HANXIN\_MAX\_LENGTH, 0x1a02b003

#### Description

Maximum data length to issue a result for Han Xin Code.

**See also**

### 10.3. DEC\_HANXIN\_MIN\_LENGTH, 0x1a02b002

#### Description

Minimum data length to issue a result for Han Xin Code.

**See also**

## 11. ITF properties

### Description

Interleaved 2 of 5

### 11.1. DEC\_I25\_BOUNDARY\_CHECK, 0x40010508

#### Description

Enables intelligent processing of quiet zones where we allow very small quiet zones, but can ensure that a read cannot occur within another bar code.

0: disabled

1: no-minimum area checked

2: requires a full specified quiet zone area to be within the image

3: requires 2X the full specified quiet zone area to be within the image

(defaults to 2)

#### See also

DEC\_SECURITY\_LEVEL

### 11.2. DEC\_I25\_ENABLED, 0x1a019001

#### Description

This property specifies whether Interleaved 2 of 5 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Interleaved 2 of 5 decoding

1: Enable Interleaved 2 of 5 decoding

Initial Value: 0

#### Example

```
/* enable Interleaved 2 of 5 decoding */
```

```
DecodeSet( DEC_I25_ENABLED, (void *) DEC_CONST_ENABLED);
```

#### See also

### 11.3. DEC\_I25\_HIGH\_DENSITY, 0x40010507

#### Description

This property improves decoding of high density Interleaved 2 of 5 barcodes. Enabling this property increases decoding time.

The property value should be set as follows:

0: Disable Interleaved 2 of 5 high density decoding improvements.

1: Enable Interleaved 2 of 5 high density decoding improvements.

Initial Value: 0

**Example**

```
/* Enable Interleaved 2 of 5 high density decoding improvements */  
DecodeSet( DEC_I25_HIGH_DENSITY, (void *) DEC_CONST_ENABLED);
```

**See also**

#### **11.4. DEC\_I25\_IMPROVE\_BOUNDS, 0x40010506**

**Description**

Controls additional processing to improve bounds.

**See also**

#### **11.5. DEC\_I25\_MAX\_LENGTH, 0x1a019003**

**Description**

Maximum data length to issue a result for Interleaved 2 of 5.

**See also**

#### **11.6. DEC\_I25\_MIN\_LENGTH, 0x1a019002**

**Description**

Minimum data length to issue a result for Interleaved 2 of 5.

**See also**

#### **11.7. DEC\_I25\_SHORT\_MARGIN, 0x40010504**

**Description**

Specifies whether substandard length margins should be allowed.

Initial value: 1

**See also**

## 11.8. DEC\_IATA25\_ENABLED, 0x1a01b001

### Description

This property specifies whether Straight 2 of 5 (with 2 bar start/stop codes) decoding is enabled during the execution of Decode. This symbology is also called: Standard 2 of 5, IATA 2 of 5, and Airline 2 of 5. "IATA" was chosen to be the short-hand for this symbology in this API.

The property value should be set as follows:

0: Disable IATA 2 of 5 decoding

1: Enable IATA 2 of 5 decoding

Initial Value: 0

### Example

```
/* enable IATA 2 of 5 decoding */
```

```
DecodeSet( DEC_IATA25_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

## 11.9. DEC\_IATA25\_MAX\_LENGTH, 0x1a01b003

### Description

Maximum data length to issue a result for IATA 2 of 5.

**See also**

## 11.10. DEC\_IATA25\_MIN\_LENGTH, 0x1a01b002

### Description

Minimum data length to issue a result for IATA 2 of 5.

**See also**

## 12. Korea Post properties

### 12.1. DEC\_KOREA\_POST\_CHECK\_DIGIT\_TRANSMIT, 0x1a100004

**Description**

Korea Post check digit transmit enable.

**See also**

### 12.2. DEC\_KOREA\_POST\_ENABLED, 0x1a100001

**Description**

This property specifies whether Korea Post decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Korea Post decoding

1: Enable Korea Post decoding

Initial Value: 0

**Example**

```
/* enable Korea Post decoding */
```

```
DecodeSet( DEC_KOREA_POST_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### 12.3. DEC\_KOREA\_POST\_MAX\_LENGTH, 0x1a100003

**Description**

Maximum data length to issue a result for Korea Post.

**See also**

### 12.4. DEC\_KOREA\_POST\_MIN\_LENGTH, 0x1a100002

**Description**

Minimum data length to issue a result for Korea Post.

**See also**

## **12.5. DEC\_KOREA\_POST\_REVERSE, 0x40013503**

### **Description**

Reverses the data prior to issuing a result for Korea Post.

0: disable

1: enable

Initial Value: 1

**See also**



## 13. Matrix 2 of 5 properties

### 13.1. DEC\_M25\_ENABLED, 0x1a01c001

#### Description

This property specifies whether Matrix 2 of 5 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Matrix 2 of 5 decoding

1: Enable Matrix 2 of 5 decoding

Initial Value: 0

#### Example

```
/* enable Matrix 2 of 5 decoding */
```

```
DecodeSet( DEC_M25_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

### 13.2. DEC\_M25\_IMPROVE\_BOUNDS, 0x40011904

#### Description

Controls additional processing to improve bounds.

See also

### 13.3. DEC\_M25\_MAX\_LENGTH, 0x1a01c003

#### Description

Maximum data length to issue a result for Matrix 2 of 5.

See also

### 13.4. DEC\_M25\_MIN\_LENGTH, 0x1a01c002

#### Description

Minimum data length to issue a result for Matrix 2 of 5.

See also

## 14. Micro PDF properties

### 14.1. DEC\_MICROPDF\_ENABLED, 0x1a025001

#### Description

This property specifies whether MicroPDF417 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable MicroPDF417 decoding

1: Enable MicroPDF417 decoding

Initial Value: 0

#### Example

```
/* enable MicroPDF417 decoding */
```

```
DecodeSet( DEC_MICROPDF_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

### 14.2. DEC\_MICROPDF\_IMPROVE\_BOUNDS, 0x40010704

#### Description

Controls additional processing to improve bounds.

See also

### 14.3. DEC\_MICROPDF\_MAX\_LENGTH, 0x1a025003

#### Description

Maximum data length to issue a result for MicroPDF417.

See also

### 14.4. DEC\_MICROPDF\_MIN\_LENGTH, 0x1a025002

#### Description

Minimum data length to issue a result for MicroPDF417.

See also

## 15. MSI Plessey properties

### 15.1. DEC\_MSIP\_SHORT\_MARGIN, 0x40011604

#### Description

This property specifies whether substandard length margins (i.e. quiet zones) should be allowed for MSI Plessey symbols during the execution of the Decode.

When this property is enabled, a substandard length quiet zone is allowed on either end (but not both ends) of a MSI Plessey symbol. Enabling this property is discouraged by Honeywell, unless absolutely necessary.

This property value is ignored if MSI Plessey decoding is not enabled using SD\_PROP\_MSIP\_ENABLED.

The property value should be set as follows:

0: Disallow short quiet zone symbols.

1: Allow short quiet zone symbols.

Property Data Type: int

Set By: Value

Initial Value: 0

**See also**

### 15.2. DEC\_MSI\_CHECK\_DIGIT\_MODE, 0x40011602

#### Description

This property specifies how MSI Plessey checksums are to be handled during the execution of the decoder.

The property value is ignored if MSI Plessey decoding is not enabled.

The property value should be set as follows:

0: Disable checksum checking.

1: Enable a single mod 10 checksum check.

2: Enable a mod 11 and a mod 10 checksum check.

3: Enable two mod 10 checksum checks.

5: Enable a single mod 10 checksum check and strip the checksum

6: Enable a mod 11 and a mod 10 checksum check and strip the checksums

7: Enable two mod 10 checksum checks and strip the checksums

**See also**

### 15.3. DEC\_MSI\_ENABLED, 0x1a021001

#### Description

This property specifies whether MSI decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable MSI decoding

1: Enable MSI decoding

Initial Value: 0

**Example**

```
/* enable MSI decoding */
```

```
DecodeSet( DEC_MSI_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

## **15.4. DEC\_MSI\_IMPROVE\_BOUNDS, 0x40011603**

**Description**

Controls additional processing to improve bounds.

**See also**

## **15.5. DEC\_MSI\_MAX\_LENGTH, 0x1a021003**

**Description**

Maximum data length to issue a result for MSI.

**See also**

## **15.6. DEC\_MSI\_MIN\_LENGTH, 0x1a021002**

**Description**

Minimum data length to issue a result for MSI.

**See also**

## **15.7. DEC\_PROP\_MSIP\_OUT\_OF\_SPEC\_SYMBOL, 0x40011605**

**Description**

This property enables enhancements to be able to read Out of Spec MSI barcodes. As the risk of misreads is increased in case of decoding Out of Spec symbols, it is recommended to only do this in case of checksums.

The property value should be set as follows:

0: Disable MSI Out of Spec Symbol Enhancements.

1: Enable MSI Enhancement to read symbols with inter character gaps.

Property Data Type: int

Set By: Value

Initial Value: 0

**See also**

## 16. Nec 2 of 5 properties

### 16.1. DEC\_NEC25\_ENABLED, 0x1a02f001

#### Description

This property specifies whether NEC 2 of 5 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable NEC 2 of 5 decoding

1: Enable NEC 2 of 5 decoding

Initial Value: 0

#### Example

```
/* enable NEC 2 of 5 decoding */
```

```
DecodeSet( DEC_NEC25_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

### 16.2. DEC\_NEC25\_IMPROVE\_BOUNDS, 0x40012204

#### Description

Controls additional processing to improve bounds.

See also

### 16.3. DEC\_NEC25\_MAX\_LENGTH, 0x1a02f003

#### Description

Maximum data length to issue a result for NEC 2 of 5.

See also

### 16.4. DEC\_NEC25\_MIN\_LENGTH, 0x1a02f002

#### Description

Minimum data length to issue a result for NEC 2 of 5.

See also

## 17. GS1 Databar properties

### Description

Formerly known as RSS

### 17.1. DEC\_RSS\_14\_ENABLED, 0x1a022001

#### Description

This property specifies whether GS1 Databar 14 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable GS1 Databar 14 decoding

1: Enable GS1 Databar 14 decoding

Initial Value: 0

#### Example

```
/* enable GS1 Databar 14 decoding */
```

```
DecodeSet( DEC_RSS_14_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

### 17.2. DEC\_RSS\_14\_IMPROVE\_BOUNDS, 0x40011302

#### Description

This property enable/disable added processing for more accurate corner output for GS1-Databar Omnidirectional. (default to 0)

#### Example

```
SI_DecodeSet( int decoder, DEC_RSS_14_IMPROVE_BOUNDS, void *data);
```

See also

### 17.3. DEC\_RSS\_EXPANDED\_ENABLED, 0x1a022003

#### Description

This property specifies whether GS1 Databar Expanded decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable GS1 Databar Expanded decoding

1: Enable GS1 Databar Expanded decoding

Initial Value: 0

#### Example

```
/* enable GS1 Databar Expanded decoding */
```

*DecodeSet( DEC\_RSS\_EXPANDED\_ENABLED, (void \*) DEC\_CONST\_ENABLED);*

**See also**

#### **17.4. DEC\_RSS\_EXPANDED\_IMPROVE\_BOUNDS, 0x40011304**

**Description**

Enable/disable added processing for more accurate corner output for GS1-Databar Expanded.  
(default to 0)

**Example**

*SI\_DecodeSet( int decoder, DEC\_RSS\_EXPANDED\_IMPROVE\_BOUNDS, void \*data);*

**See also**

#### **17.5. DEC\_RSS\_EXPANDED\_MAX\_LENGTH, 0x1a022005**

**Description**

Maximum data length to issue a result for GS1-Databar Expanded.

**See also**

#### **17.6. DEC\_RSS\_EXPANDED\_MIN\_LENGTH, 0x1a022004**

**Description**

Minimum data length to issue a result for GS1-Databar Expanded.

**See also**

#### **17.7. DEC\_RSS\_LIMITED\_ENABLED, 0x1a022002**

**Description**

This property specifies whether GS1 Databar Limited decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable GS1 Databar Limited decoding

1: Enable GS1 Databar Limited decoding

Initial Value: 0



**Example**

```
/* enable GS1 Databar Limited decoding */
```

```
DecodeSet( DEC_RSS_LIMITED_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

## **17.8. DEC\_RSS\_LIMITED\_IMPROVE\_BOUNDS, 0x40011303**

**Description**

Enable/disable added processing for more accurate corner output for GS1-Databar Limited.  
(default to 0)

**Example**

```
SI_DecodeSet( int decoder, DEC_RSS_LIMITED_IMPROVE_BOUNDS, void *data)
```

**See also**

## 18. Standard 2 of 5 properties

### 18.1. DEC\_IATA25\_IMPROVE\_BOUNDS, 0x40011505

#### Description

Controls additional processing to improve bounds.

See also

### 18.2. DEC\_S25\_ENABLED, 0x1a01a001

#### Description

This property specifies whether Straight 2 of 5 (with 3 bar start/stop codes) decoding is enabled during the execution of Decode.

This symbology is also called: Industrial 2 of 5, Code 2 of 5, and Discrete 2 of 5.

"S" for "Straight" was chosen to be the short-hand for this symbology in this API.

The property value should be set as follows:

0: Disable S 2 of 5 decoding

1: Enable S 2 of 5 decoding

Initial Value: 0

#### Example

```
/* enable S 2 of 5 decoding */
```

```
DecodeSet( DEC_S25_ENABLED, (void *) DEC_CONST_ENABLED);
```

See also

### 18.3. DEC\_S25\_IMPROVE\_BOUNDS, 0x40011506

#### Description

Controls additional processing to improve bounds.

See also

### 18.4. DEC\_S25\_MAX\_LENGTH, 0x1a01a003

#### Description

Maximum data length to issue a result for Straight 2 of 5.

See also

## **18.5. DEC\_S25\_MIN\_LENGTH, 0x1a01a002**

### **Description**

Minimum data length to issue a result for Straight 2 of 5.

### **See also**

## 19. Telepen properties

### 19.1. DEC\_TELEPEN\_ENABLED, 0x1a020001

#### Description

This property specifies whether Telepen decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Telepen decoding

1: Enable Telepen decoding

Initial Value: 0

#### Example

```
/* enable Telepen decoding */
```

```
DecodeSet( DEC_TELEPEN_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### 19.2. DEC\_TELEPEN\_IMPROVE\_BOUNDS, 0x40012103

#### Description

Controls additional processing to improve bounds.

**See also**

### 19.3. DEC\_TELEPEN\_MAX\_LENGTH, 0x1a020003

#### Description

Maximum data length to issue a result for Telepen.

**See also**

### 19.4. DEC\_TELEPEN\_MIN\_LENGTH, 0x1a020002

#### Description

Minimum data length to issue a result for Telepen.

**See also**

## **19.5. DEC\_TELEPEN\_OLD\_STYLE, 0x40012102**

### **Description**

Converts data using old algorithm prior to issuing a result for Telepen.

### **See also**

## 20. Trioptic properties

### 20.1. DEC\_TRIOPTIC\_ENABLED, 0x1a018001

**Description**

This property specifies whether Trioptic decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Trioptic decoding

1: Enable Trioptic decoding

Trioptic is a variation of standard Code 39 that uses a different start/stop character and is a fixed length of 6 data codewords (8 codewords overall counting the start and stop characters).

The Decoder Control Logic will perform the reordering of codewords before calling the result callback.

For example, the data content was 123456, it is output as 456123.

Initial Value: 0

**Example**

```
/* enable Trioptic decoding */
```

```
DecodeSet( DEC_TRIOPTIC_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### 20.2. DEC\_TRIOPTIC\_SHORT\_MARGIN, 0x40010308

**Description**

Specifies whether substandard length margins should be allowed.

Initial value: 1

**See also**

## 21. UPC/EAN properties

### 21.1. DEC\_2CHAR\_ADDENDA\_REDUCED\_SEPARATION , 0x0a010008

#### Description

Allow short quiet zone for a 2-digit addenda.

The property value should be set as follows:

0: Disable.

1: Enable.

Property Data Type: int

Set By: Value

Initial Value: 0

#### Example

```
SD_Set(Handle, DEC_2CHAR_ADDENDA_REDUCED_SEPARATION, (void *) SD_CONST_ENABLED);
```

#### See also

### 21.2. DEC\_ADD\_SEARCH\_TIME\_ADDENDA, 0x1a003004

#### Description

Minimum additional time to decode a UPC addenda.

#### See also

### 21.3. DEC\_EAN13\_290\_ADDENDA\_REQUIRED, 0x1a013009

#### Description

This property specifies whether a 5 chars addenda is required for EAN 13 starting with "290".

When enabled, in order to obtain a result, a 5 chars addenda must be decodable in the same image if the EAN13 starts with "290".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

0: Disable

1: Enable

Initial Value: 0

#### Example

```
/* enable 5 chars addenda for EAN13 codes starting with 290 */
```

*DecodeSet( DEC\_EAN13\_290\_ADDENDA\_REQUIRED, (void \*)1 );*

**See also**

DEC\_EAN13\_378\_ADDENDA\_REQUIRED  
DEC\_EAN13\_414\_ADDENDA\_REQUIRED  
DEC\_EAN13\_434\_ADDENDA\_REQUIRED  
DEC\_EAN13\_491\_ADDENDA\_REQUIRED  
DEC\_EAN13\_977\_ADDENDA\_REQUIRED  
DEC\_EAN13\_978\_ADDENDA\_REQUIRED  
DEC\_EAN13\_979\_ADDENDA\_REQUIRED

## **21.4. DEC\_EAN13\_2CHAR\_ADDENDA\_ENABLED, 0x1a013003**

**Description**

EAN-13 2-character addenda enable.

**See also**

## **21.5. DEC\_EAN13\_378\_ADDENDA\_REQUIRED, 0x1a01300a**

**Description**

This property specifies whether an addenda is required for EAN 13 starting with "378" or "379".

When enabled, in order to obtain a result, the addenda specified must be decodable in the same image if the EAN13 starts with "378" or "379".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

- 0: Disable
- 1: Requires 2 chars addenda
- 2: Requires 5 chars addenda
- 3: Requires 2 or 5 chars addenda

Initial Value: 0

**Example**

```
/* enable 5 chars addenda for EAN13 codes starting with 378 or 379 */  
DecodeSet( DEC_EAN13_378_ADDENDA_REQUIRED, (void *)2 );
```

**See also**

DEC\_EAN13\_290\_ADDENDA\_REQUIRED  
DEC\_EAN13\_414\_ADDENDA\_REQUIRED  
DEC\_EAN13\_434\_ADDENDA\_REQUIRED  
DEC\_EAN13\_491\_ADDENDA\_REQUIRED  
DEC\_EAN13\_977\_ADDENDA\_REQUIRED



DEC\_EAN13\_978\_ADDENDA\_REQUIRED

DEC\_EAN13\_979\_ADDENDA\_REQUIRED

## 21.6. DEC\_EAN13\_414\_ADDENDA\_REQUIRED, 0x1a01300b

### Description

This property specifies whether an addenda is required for EAN 13 starting with "414" or "419".

When enabled, in order to obtain a result, the addenda specified must be decodable in the same image if the EAN13 starts with "414" or "419".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

0: Disable

1: Requires 2 chars addenda

2: Requires 5 chars addenda

3: Requires 2 or 5 chars addenda

Initial Value: 0

### Example

*/\* enable 5 chars addenda for EAN13 codes starting with 414 or 419 \*/*

*DecodeSet( DEC\_EAN13\_414\_ADDENDA\_REQUIRED, (void \*)2 );*

### See also

DEC\_EAN13\_290\_ADDENDA\_REQUIRED

DEC\_EAN13\_378\_ADDENDA\_REQUIRED

DEC\_EAN13\_434\_ADDENDA\_REQUIRED

DEC\_EAN13\_491\_ADDENDA\_REQUIRED

DEC\_EAN13\_977\_ADDENDA\_REQUIRED

DEC\_EAN13\_978\_ADDENDA\_REQUIRED

DEC\_EAN13\_979\_ADDENDA\_REQUIRED

## 21.7. DEC\_EAN13\_434\_ADDENDA\_REQUIRED, 0x1a01300c

### Description

This property specifies whether an addenda is required for EAN 13 starting with "434" or "439".

When enabled, in order to obtain a result, the addenda specified must be decodable in the same image if the EAN13 starts with "434" or "439".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

0: Disable

1: Requires 2 chars addenda

- 2: Requires 5 chars addenda
- 3: Requires 2 or 5 chars addenda

Initial Value: 0

**Example**

```
/* enable 5 chars addenda for EAN13 codes starting with 434 or 439 */  
DecodeSet( DEC_EAN13_434_ADDENDA_REQUIRED, (void *)2 );
```

**See also**

DEC\_EAN13\_290\_ADDENDA\_REQUIRED  
DEC\_EAN13\_378\_ADDENDA\_REQUIRED  
DEC\_EAN13\_414\_ADDENDA\_REQUIRED  
DEC\_EAN13\_491\_ADDENDA\_REQUIRED  
DEC\_EAN13\_977\_ADDENDA\_REQUIRED  
DEC\_EAN13\_978\_ADDENDA\_REQUIRED  
DEC\_EAN13\_979\_ADDENDA\_REQUIRED

## **21.8. DEC\_EAN13\_491\_ADDENDA\_REQUIRED, 0x1a013010**

**Description**

This property specifies whether an addenda is required for EAN 13 starting with "491".

When enabled, in order to obtain a result, the addenda specified must be decodable in the same image if the EAN13 starts with "491".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

- 0: Disable
- 1: Requires 2 chars addenda
- 2: Requires 5 chars addenda
- 3: Requires 2 or 5 chars addenda

Initial Value: 0

**Example**

```
/* enable 5 chars addenda for EAN13 codes starting with 491 */  
DecodeSet( DEC_EAN13_491_ADDENDA_REQUIRED, (void *)2 );
```

**See also**

DEC\_EAN13\_290\_ADDENDA\_REQUIRED  
DEC\_EAN13\_378\_ADDENDA\_REQUIRED  
DEC\_EAN13\_414\_ADDENDA\_REQUIRED  
DEC\_EAN13\_434\_ADDENDA\_REQUIRED  
DEC\_EAN13\_977\_ADDENDA\_REQUIRED  
DEC\_EAN13\_978\_ADDENDA\_REQUIRED  
DEC\_EAN13\_979\_ADDENDA\_REQUIRED

## 21.9. DEC\_EAN13\_5CHAR\_ADDENDA\_ENABLED, 0x1a013004

### Description

EAN-13 5-character addenda enable.

### See also

## 21.10. DEC\_EAN13\_977\_ADDENDA\_REQUIRED, 0x1a01300d

### Description

This property specifies whether a 2 chars addenda is required for EAN 13 starting with "977".

When enabled, in order to obtain a result, a 2 chars addenda must be decodable in the same image if the EAN13 starts with "977".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

0: Disable

1: Enable

Initial Value: 0

### Example

```
/* enable 2 chars addenda for EAN13 codes starting with 977 */  
DecodeSet( DEC_EAN13_977_ADDENDA_REQUIRED, (void *)1 );
```

### See also

DEC\_EAN13\_290\_ADDENDA\_REQUIRED

DEC\_EAN13\_378\_ADDENDA\_REQUIRED

DEC\_EAN13\_414\_ADDENDA\_REQUIRED

DEC\_EAN13\_434\_ADDENDA\_REQUIRED

DEC\_EAN13\_491\_ADDENDA\_REQUIRED

DEC\_EAN13\_978\_ADDENDA\_REQUIRED

DEC\_EAN13\_979\_ADDENDA\_REQUIRED

## 21.11. DEC\_EAN13\_978\_ADDENDA\_REQUIRED, 0x1a01300e

### Description

This property specifies whether a 5 chars addenda is required for EAN 13 starting with "978".

When enabled, in order to obtain a result, a 5 chars addenda must be decodable in the same image if the EAN13 starts with "978".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

0: Disable

1: Enable

Initial Value: 0

#### Example

```
/* enable 5 chars addenda for EAN13 codes starting with 978 */
DecodeSet( DEC_EAN13_978_ADDENDA_REQUIRED, (void *)1 );
```

#### See also

DEC\_EAN13\_290\_ADDENDA\_REQUIRED

DEC\_EAN13\_378\_ADDENDA\_REQUIRED

DEC\_EAN13\_414\_ADDENDA\_REQUIRED

DEC\_EAN13\_434\_ADDENDA\_REQUIRED

DEC\_EAN13\_491\_ADDENDA\_REQUIRED

DEC\_EAN13\_977\_ADDENDA\_REQUIRED

DEC\_EAN13\_979\_ADDENDA\_REQUIRED

## 21.12. DEC\_EAN13\_979\_ADDENDA\_REQUIRED, 0x1a01300f

### Description

This property specifies whether a 5 chars addenda is required for EAN 13 starting with "979".

When enabled, in order to obtain a result, a 5 chars addenda must be decodable in the same image if the EAN13 starts with "979".

This property takes precedence on other addenda requirement properties.

The property value should be set as follows:

0: Disable

1: Enable

Initial Value: 0

#### Example

```
/* enable 5 chars addenda for EAN13 codes starting with 979 */
DecodeSet( DEC_EAN13_979_ADDENDA_REQUIRED, (void *)1 );
```

#### See also

DEC\_EAN13\_290\_ADDENDA\_REQUIRED

DEC\_EAN13\_378\_ADDENDA\_REQUIRED

DEC\_EAN13\_414\_ADDENDA\_REQUIRED

DEC\_EAN13\_434\_ADDENDA\_REQUIRED

DEC\_EAN13\_491\_ADDENDA\_REQUIRED

DEC\_EAN13\_977\_ADDENDA\_REQUIRED

DEC\_EAN13\_978\_ADDENDA\_REQUIRED

### **21.13. DEC\_EAN13\_ADDENDA\_REQUIRED, 0x1a013005**

#### **Description**

Requires reading an addenda for EAN-13 to issue a result.

**See also**

### **21.14. DEC\_EAN13\_ADDENDA\_SEPARATOR, 0x1a013006**

#### **Description**

Adds a space between main code and addenda for EAN-13.

**See also**

### **21.15. DEC\_EAN13\_ENABLED, 0x1a013001**

#### **Description**

This property specifies whether EAN 13 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable EAN 13 decoding

1: Enable EAN 13 decoding

Initial Value: 1

#### **Example**

```
/* enable EAN 13 decoding */
```

```
DecodeSet( DEC_EAN13_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### **21.16. DEC\_EAN13\_ISBN\_ENABLED, 0x1a013007**

#### **Description**

EAN-13 ISBN format handling enable.

**See also**

### **21.17. DEC\_EAN13\_ISSN\_ENABLED, 0x1a013011**

**Description**

EAN-13 ISSN format handling enable.

**See also**

DEC\_EAN13\_ISSN\_SEPARATOR

DEC\_EAN13\_ENABLED

**21.18. DEC\_EAN13\_ISSN\_SEPARATOR, 0x1a013012**

**Description**

Inserts ISSN separator.

**See also**

DEC\_EAN13\_ISSN\_ENABLED

DEC\_EAN13\_ENABLED

**21.19. DEC\_EAN8\_2CHAR\_ADDENDA\_ENABLED, 0x1a012003**

**Description**

EAN-8 2-character addenda enable.

**See also**

**21.20. DEC\_EAN8\_5CHAR\_ADDENDA\_ENABLED, 0x1a012004**

**Description**

EAN-8 5-character addenda enable.

**See also**

**21.21. DEC\_EAN8\_ADDENDA\_REQUIRED, 0x1a012005**

**Description**

Requires reading an addenda for EAN-8 to issue a result.

**See also**

## **21.22. DEC\_EAN8\_ADDENDA\_SEPARATOR, 0x1a012006**

### **Description**

Adds a space between main code and addenda for EAN-8.

**See also**

## **21.23. DEC\_EAN8\_CHECK\_DIGIT\_TRANSMIT, 0x1a012002**

### **Description**

EAN-8 check digit transmit enable.

**See also**

## **21.24. DEC\_EAN8\_ENABLED, 0x1a012001**

### **Description**

This property specifies whether EAN 8 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable EAN 8 decoding

1: Enable EAN 8 decoding

Initial Value: 1

### **Example**

```
/* enable EAN 8 decoding */
```

```
DecodeSet( DEC_EAN8_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

## **21.25. DEC\_UPCA\_2CHAR\_ADDENDA\_ENABLED, 0x1a010004**

### **Description**

UPC-A 2-character addenda enable.

**See also**

## **21.26. DEC\_UPCA\_5CHAR\_ADDENDA\_ENABLED, 0x1a010005**

**Description**

UPC-A 5-character addenda enable.

**See also**

**21.27. DEC\_UPCA\_ADDENDA\_REQUIRED, 0x1a010006**

**Description**

Requires reading an addenda for UPC-A to issue a result.

**See also**

**21.28. DEC\_UPCA\_ADDENDA\_SEPARATOR, 0x1a010007**

**Description**

Added a space between main code and addenda for UPC-A.

**See also**

**21.29. DEC\_UPCA\_CHECK\_DIGIT\_TRANSMIT, 0x1a010002**

**Description**

UPC-A check digit transmit enable.

**See also**

**21.30. DEC\_UPCA\_ENABLE, 0x1a010011**

**Description**

This property specifies whether UPCA decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable UPCA decoding

1: Enable UPCA decoding

Initial Value: 1

**See also**

DEC\_UPCA\_TRANSLATE\_TO\_EAN13



### 21.31. DEC\_UPCA\_NUMBER\_SYSTEM\_TRANSMIT, 0x1a010003

#### Description

UPC-A number system transmit enable.

#### See also

### 21.32. DEC\_UPCA\_TRANSLATE\_TO\_EAN13, 0x1a010001

#### Description

This property specifies whether UPC-A are translated to EAN13.

The property value should be set as follows:

0: Disable UPC-A to EAN13 translation

1: Enable UPC-A to EAN13 translation

Initial Value: 0

#### Example

```
/* enable UPC-A to EAN13 translation */
```

```
DecodeSet( DEC_UPCA_TRANSLATE_TO_EAN13, (void *) 0x01);
```

#### See also

DEC\_UPCA\_ENABLE

### 21.33. DEC\_UPCE0\_ENABLED, 0x1a011001

#### Description

This property specifies whether UPC-E0 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable UPC-E0 decoding

1: Enable UPC-E0 decoding

Initial Value: 1

#### Example

```
/* enable UPC-E0 decoding */
```

```
DecodeSet( DEC_UPCE0_ENABLED, (void *) DEC_CONST_ENABLED);
```

#### See also

### 21.34. DEC\_UPCE1\_ENABLED, 0x1a011002

**Description**

This property specifies whether UPC-E1 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable UPC-E1 decoding

1: Enable UPC-E1 decoding

Initial Value: 0

**Example**

```
/* enable UPC-E1 decoding */
```

```
DecodeSet( DEC_UPCE1_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

**21.35. DEC\_UPCE\_2CHAR\_ADDENDA\_ENABLED, 0x1a011006****Description**

UPC-E 2-character addenda enable.

**See also**

**21.36. DEC\_UPCE\_5CHAR\_ADDENDA\_ENABLED, 0x1a011007****Description**

UPC-E 5-character addenda enable.

**See also**

**21.37. DEC\_UPCE\_ADDENDA\_REQUIRED, 0x1a011008****Description**

Requires reading an addenda for UPC-E to issue a result.

**See also**

**21.38. DEC\_UPCE\_ADDENDA\_SEPARATOR, 0x1a011009****Description**

Adds a space between main code and addenda for UPC-E.

**See also**

#### **21.39. DEC\_UPCE\_CHECK\_DIGIT\_TRANSMIT, 0x1a011004**

**Description**

UPC-E check digit transmit enable.

**See also**

#### **21.40. DEC\_UPCE\_EXPAND, 0x1a011003**

**Description**

UPC-E expand enable.

**See also**

#### **21.41. DEC\_UPCE\_NUMBER\_SYSTEM\_TRANSMIT, 0x1a011005**

**Description**

UPC-E number system transmit enable.

**See also**

#### **21.42. DEC\_UPC\_IMPROVE\_BOUNDS, 0x40011006**

**Description**

Controls additional processing to improve bounds.

**See also**

#### **21.43. DEC\_UPC\_OUT\_OF\_SPEC\_SYMBOL, 0x40011009**

**Description**

Enable decoding out-of-spec UPC. The property is a bitmap and currently only bit0 (1 << 0) is supported. When

set, this bit allows decoding UPC in which the variation in width between adjacent codewords is larger than usual.

**See also**

#### **21.44. DEC\_UPC\_SHORT\_MARGIN, 0x40011004**

**Description**

Specifies whether substandard length margins should be allowed.

Initial value: 1

**See also**

## 22. OCR properties

### 22.1. DEC\_OCR\_ACTIVE\_TEMPLATES, 0x1b02d003

**Description**

Specifies which templates are active for decoding an OCR string.

**See also**

### 22.2. DEC\_OCR\_BUSY\_BACKGROUND, 0x40012308

**Description**

Controls algorithms that improve decoding when the background field on which OCR tests is printed is not uniform.

0: Disable

1: Enable

Default: 0

**See also**

### 22.3. DEC\_OCR\_MODE, 0x1a02d001

**Description**

Optical Character Recognition mode.

0: Disable

1: Normal video

2: Reverse Video

3: Both Video

Default: 0

**See also**

### 22.4. DEC\_OCR\_PASSPORT\_IGNORE\_CHECKSUM, 0x1b02d006

**Description**

Specifies whether ICAO checksum calculations are preformed.

0: Perform checksum calculations and suppress OCR output for rows with bad checksums.

1: Checksum calculations are not performed. OCR output with bad checksums may be issued.

Default: 0

**See also**

## **22.5. DEC\_OCR\_TEMPLATE, 0x9a02d002**

### **Description**

Template for reading Optical Characters.

**See also**

## 23. Image properties

### 23.1. DEC\_IMAGE\_BINNING\_SIZE, 0x40004012

#### Description

This property enables internal binning of the image. It can be used to decrease the average decode time. The disadvantage is that symbols with small features may not be decoded.

The property value should be set as follows:

- 1: Do not perform any image binning
- 2: Bin every 2x2 area in the image

Initial Value: 1

**See also**

### 23.2. DEC\_IMAGE\_MIRRORED, 0x40004003

#### Description

Sets if the image is mirrored.

**See also**

### 23.3. DEC\_IMAGE\_ROTATION, 0x1a001015

#### Description

Sets the orientation of the image.

**See also**

### 23.4. DEC\_WINDOW\_BOTTOM, 0x1a00100d

#### Description

Bottom border of window in percentage of the image height.

**See also**

DEC\_WINDOW\_MODE  
DEC\_WINDOW\_TOP  
DEC\_WINDOW\_LEFT  
DEC\_WINDOW\_RIGHT

**23.5. DEC\_WINDOW\_LEFT, 0x1a00100e****Description**

Left border of window in percentage of the image width.

**See also**

DEC\_WINDOW\_MODE

DEC\_WINDOW\_TOP

DEC\_WINDOW\_BOTTOM

DEC\_WINDOW\_RIGHT

**23.6. DEC\_WINDOW\_MODE, 0x1a00100b****Description**

Windowing or centering mode.

This property restricts the region of the image in which the decoder will decode bar codes. In order to make use of this property, the user must define a windowing region by setting the following related properties:

- DEC\_WINDOW\_TOP
- DEC\_WINDOW\_BOTTOM
- DEC\_WINDOW\_LEFT
- DEC\_WINDOW\_RIGHT

These are referred to as the window bounds properties.

The DEC\_WINDOW\_MODE property value should be set to one of the following primary modes:

- 0: Windowing mode disabled
- 1: Search-center-relative windowing mode
- 2: Image-relative windowing mode
- 3: Clipped windowing mode

**# Windows Mode Details****## Mode 0: Windowing mode disabled**

In mode 0, the decoder processes the entire supplied image frame, and any bar codes found are reported via the result callback.

**## Mode 1: Search-center-relative windowing mode**

In mode 1, the decoder defines a region within the image based on the window bounds properties. Any bar code the decoder detects that intersects with this region is reported via the result callback. Bar codes that do not intersect are ignored. The region is sized relative to the image dimensions and centered about the search center, which is defined by the user-specified DEC\_IMAGE\_CENTER\_X and DEC\_IMAGE\_CENTER\_Y properties. Note that to use this mode, the window bounds properties must be set such that they span 50% in both dimensions. Otherwise, the decoder will operate as if it were set to mode 2. For example, the decoder can operate in mode 1 when DEC\_WINDOW\_TOP = 40 and DEC\_WINDOW\_BOTTOM = 60 because  $40 < 50 < 60$ . However, it fallback to the mode 2 behavior if DEC\_WINDOW\_TOP = 30 and DEC\_WINDOW\_BOTTOM = 40. The same constraint applies to DEC\_WINDOW\_LEFT and DEC\_WINDOW\_RIGHT.

**## Mode 2: Image-relative windowing mode**

Mode 2 operates identically to mode 1, except the region is not centered about the search center.



#### ## Mode 3: Clipped windowing mode

In mode 3, the decoder defines the region in the same manner as in mode 2 (i.e. image-relative), but it does not process any portions of the image that lie outside of the region. Mode 3 is the fastest windowing mode, but the decoder will only read bar codes that lie entirely within the region.

#### See also

DEC\_WINDOW\_TOP  
DEC\_WINDOW\_BOTTOM  
DEC\_WINDOW\_LEFT  
DEC\_WINDOW\_RIGHT

### 23.7. DEC\_WINDOW\_RIGHT, 0x1a00100f

#### Description

Right border of window in percentage of the image height.

#### See also

DEC\_WINDOW\_MODE  
DEC\_WINDOW\_TOP  
DEC\_WINDOW\_BOTTOM  
DEC\_WINDOW\_LEFT

### 23.8. DEC\_WINDOW\_TOP, 0x1a00100c

#### Description

Top border of window in percentage of the image height.

#### See also

DEC\_WINDOW\_MODE  
DEC\_WINDOW\_BOTTOM  
DEC\_WINDOW\_LEFT  
DEC\_WINDOW\_RIGHT

## 24. Result properties

### 24.1. DEC\_RESULT\_APPEND\_TYPE, 0x1a001025

#### Description

Gets the Append type of the result.

The possible types are:

0x00001: Code 39  
 0x00002: Coupon Code  
 0x00004: Code 93  
 0x00008: MacroPDF  
 0x00010: Code 128  
 0x00020: Aztec code  
 0x00040: QR code  
 0x00080: DataMatrix  
 0x00100: GridMatrix  
 0x00200: UCC Composite  
 0x00400: Codabar  
 0x00800: ISBT  
 0x01000: UPC Composite  
 0x02000: OCR  
 0x04000: OCR User  
 0x08000: KIX Reverse  
 0x10000: HIBC

#### See also

### 24.2. DEC\_RESULT\_LINK\_FLAG, 0x1a001028

#### Description

Gets the Link flag of the result.

When the 1D and 2D components of a composite code are combined, the link flag reported is the one from the 2D component to help to distinguish the type of the composite code read.

0: None  
 11: Code 128 linked  
 12: GS1-Databar linked  
 13: Code 39 linked  
 21: Composite CC-A linked  
 22: Composite CC-B linked  
 23: Composite CC-C linked  
 24: MicroPDF linked

#### See also

## 25. Miscellaneous properties

### 25.1. DEC\_ADD\_SEARCH\_TIME\_CONCAT, 0x1a003005

**Description**

Minimum additional time to decode certain appends.

**See also**

### 25.2. DEC\_BLACK\_LEVEL, 0x40005013

**Description**

Average black level of the background on which scanned objects appear.

**See also**

### 25.3. DEC\_CYCLING\_FINDER, 0x40100006

**Description**

It configures how the finder scans will be done on decoder.

For linear code, when cycling finder is enabled the decoder focuses its search for bar codes on the center of the image. This is appropriate for handheld readers and applications where the bar code to be read is expected to always appear near the center of the image. When this property is disabled, the decoder performs a uniform search over the entire image.

This property has no effect if DEC\_PASS\_THROUGH is activated.

When this property is enabled, bar codes located far from the image center may not be decoded.

The property values are:

0: disable

1: enable

default value: 1

**Example**

```
/* enable uniform search over the entire image */  
DecodeSet( DEC_CYCLING_FINDER, (void *) 0x00);
```

**See also**

DEC\_PASS\_THROUGH

**25.4. DEC\_DECODE\_VIGOR, 0x1a001001****Description**

Algorithms for more vigorous searching decoding. DEC\_DECODE\_VIGOR controls the ID property SD\_PROP\_DM\_ENHANCED\_DAMAGE\_MODE. It is not a direct-mapped property. DEC\_DECODE\_VIGOR has 3 values, 0, 1, and 2. But values 0 and 2 behave identically. Value 1 enables the ID property. The other values disable it.

**Example**

```
ID_DeclareSet( int decoder, DEC_DECODE_VIGOR, void *data );
```

**See also****25.5. DEC\_DEFER\_RESULTS, 0x1a002004****Description**

When set, the decoder is buffering the decodes, the callback function to output the results will be called once when one of the following criteria is true:

- the decoder has exhausted the image
- the decoder is stopped from the outside
- the decoder has buffered 10 results

When disable, the callback function to output a result is called whenever a result is available.

**See also****25.6. DEC\_DPM\_ENABLED, 0x40012903****Description**

This property specifies whether DPM decoding is enabled during the execution of Decode.

The property value should be set as follows:

- 0: Disable DPM decoding
- 1: Dotpeen DPM decoding
- 2: Reflective DPM decoding

Initial Value: 0

**Example**

```
/* enable dotpeen DPM decoding */
DecodeSet( DEC_DPM_ENABLED, (void *) 0x01);
```

**See also**

DEC\_DPM\_REFLECTIVE\_SIZE  
DEC\_ROI\_MODE

**25.7. DEC\_DPM\_REFLECTIVE\_SIZE, 0x40012904****Description**

This property can improve decoding of reflective DPM symbols when the size is small.

The property is ignored if DEC\_DPM\_ENABLED is not set to 2 (Reflective DPM decoding).

The property value should be set as follows:

0: Normal reflective DPM size.

1: Small reflective DPM size.

Initial Value: 0

**Example**

```
/* enable small reflective DPM decoding */
```

```
DecodeSet( DEC_DPM_REFLECTIVE_SIZE, (void *) 0x01);
```

**See also**

DEC\_DPM\_ENABLED

**25.8. DEC\_ECI\_HANDLING, 0x1a002003****Description**

ECI codeword handling in result data.

**See also****25.9. DEC\_EDGE\_DETECTOR, 0x40100003****Description**

This property allows to increase depth of field for linear codes with a tradeoff of longer computation time.

0: Disable enhanced depth of field

1: Enable enhanced depth of field

Property Data Type: int

Set By: Value

Initial Value: 1

**Example**

```
/* enable increased depth of field */
```

```
DecodeSet( DEC_EDGE_DETECTOR, (void *) 0x01);
```

**See also**

## **25.10. DEC\_FLD\_DECODE\_ATTEMPTS, 0xfd012009**

### **Description**

Maximum number of decode attempts for FLD. Multiple decoding attempts on a single image is now supported. This may be allowed by setting the DEC\_FLD\_DECODE\_ATTEMPTS property greater than 1.

**See also**

## **25.11. DEC\_FLD\_ENHANCED\_DOOF, 0xFD012011**

### **Description**

This property enables enhanced depth-of-field processing for linear symbologies in Fast Linear Decoder (FLD). Currently, only UPC-A and EAN-13 are affected. More symbologies may be added in future versions of the decoder. The far Vesta filter must be enabled via DEC\_FLD\_VESTA\_FILTERS for this property to have any effect.

The property value should be set as follows:

0: enhanced DoF disabled

1: enhanced DoF enabled

Initial Value: 0

**See also**

## **25.12. DEC\_FLD\_ROLLING\_FILTERS, 0xfd012004**

### **Description**

This property specifies whether all the decode filters in the FLD decoder are applied on the same image or on several images.

The property value should be set as follows:

0: on the same image.

1: on two consecutive images.

Initial Value: 1

Note: If all the filters are applied on a same image, the time spent in FLD is higher.

**See also**

## **25.13. DEC\_FLD\_SEARCH\_AREA, 0xfd012002**

### **Description**

This property specifies the area analysed by the FLD decoder to find a barcode.

The property value should be set as follows:

0: center of the image.

1: full image.

Initial Value: 0

**See also**

#### **25.14. DEC\_FLD\_SEARCH\_HEIGHT, 0xfd012007**

**Description**

Override the search area height for FLD.

Height %

Property Data Tpe: int

Set By: value

**See also**

DEC\_FLD\_SEARCH\_AREA

#### **25.15. DEC\_FLD\_SEARCH\_SPACING, 0xfd012008**

**Description**

Override the search spacing for FLD.

**See also**

DEC\_FLD\_SEARCH\_AREA

#### **25.16. DEC\_FLD\_SEARCH\_WIDTH, 0xfd012006**

**Description**

Override the search area width for FLD.

Widht %

Property Data Tpe: int

Set By: value

**See also**

DEC\_FLD\_SEARCH\_AREA

#### **25.17. DEC\_FLD\_SUMMING, 0xfd012003**

**Description**

This property specifies the amount of summing used by the FLD decoder to decode poor printed barcodes.

The property value should be set as follows:

0: no summing.

1: summing of 3 pixels.

2: summing of 5 pixels.

Initial value is 0.

Note: The higher is the property value, the higher is the time spent in the FLD decoder.

**See also****25.18. DEC\_GENERAL\_IMPROVEMENTS, 0x40005011****Description**

This property specifies whether certain improvements should be enabled.

The property value is a bit field defined as follow:

b0: Enable improved positioning of the bounds for POSTNET and PLANET code symbols.

The values returned by SD\_PROP\_RESULT\_BOUNDS will indicate the starting and ending positions of the symbol with greatly improved accuracy.

b1: Enable improved symbol locating algorithms, especially at lower sampling density.

b2: For Code 128, enable improved compliance with sections 2.3.4 ("Special Characters") and 2.7 ("Transmitted Data") of the Uniform Symbology Specification – Code 128 (June 1993 edition).

Initial value: 0x00

**See also****25.19. DEC\_GET\_ENERGY, 0x1a001013****Description**

Returns energy of image.

**See also****25.20. DEC\_ID\_MULTITHREADING, 0x40100015****Description**

This property specifies whether the decoding should be performed on background threads. When enabled, the



execution of the decoder will take place on multiple threads. Only a single callback function may be called at a time, but the execution of the callback properties will be performed from the context of the background threads.

The property values are:

0: disable

1: enable

Initial Value: 0

**See also**

## **25.21. DEC\_ISSUE\_IDENTICAL\_SYMBOLS, 0x40005004**

### **Description**

Will issue callback for each identical symbol in image.

**See also**

## **25.22. DEC\_LINEAR\_DECODE\_POLARITY, 0x1a00102b**

### **Description**

This property specifies whether normal or inverse decoding for linear symbologies is enabled during the execution of Decode. This property only works if DEC\_VIDEO\_REVERSE\_ENABLED is set to 0. It is recommended to use this property instead of DEC\_VIDEO\_REVERSE\_ENABLED.

0: Decode only normal video for 1D codes

1: Decode only inverse video for 1D codes

2: Decode both, normal and inverse video for 1D codes

### **Example**

```
/* enable normal video for 1D decoding */
```

```
DecodeSet( DEC_LINEAR_DECODE_POLARITY, (void *) 0 );
```

**See also**

DEC\_VIDEO\_REVERSE\_ENABLED

## **25.23. DEC\_LOW\_ASPECT\_RATIO, 0x40005005**

### **Description**

Controls low aspect ratio searching for 1D symbols (including PDF417).

Initial value: 1

**See also**

#### **25.24. DEC\_LOW\_CONTRAST\_IMPROVEMENTS, 0x40005006**

**Description**

Controls low contrast searching.

Initial value: 1

**See also**

#### **25.25. DEC\_MIN\_PPM\_1D, 0x1a003008**

**Description**

If the value of the property is not 0, a 1D barcode can be decoded only if the resolution in the image of its narrow bars is higher than the property value. The measurement unit is 1/10 of pixel.

This property is only available for: Code 39, Codabar, Code 128, EAN/UPC, Interleaved 2 of 5, Code 11, Code 93, MSI, PDF/microPDF and GS1 Databar.

Initial Value: 0 (No resolution check)

**See also**

DEC\_MIN\_PPM\_2D

#### **25.26. DEC\_MIN\_PPM\_2D, 0x1a003009**

**Description**

If the value of the property is not 0, a 2D barcode can be decoded only if the resolution in the image of its modules is higher than the property value. The measurement unit is 1/10 of pixel.

This property is only available for: Aztec, Datamatrix and QR code.

Initial Value: 0 (No resolution check)

**See also**

DEC\_MIN\_PPM\_1D

#### **25.27. DEC\_PASS\_THROUGH, 0x40005016**

**Description**

Controls time spent searching and decoding.

**See also**

## **25.28. DEC\_ROI\_MODE, 0x40008015**

### **Description**

When set, only the region(s) of interest is(are) processed by the decoder.

0: Disable(default value). The entire original image is sent to the Decoder.

1: Standard

- Use the aimer position to weight activity.
- Activity calculated on the row and the column in the middle of each cell.
- The ROI window may not include the aimer.
- This is the default mode recommended for hand held scanner when used in Manual Trigger mode.

2: Standard, aimer centered.

- Activity calculated on the row and the column in the middle of each cell.
- The ROI window will always include the aimer.
- This is the default mode recommended for Andaman.

3: DPM, aimer centered.

- Activity calculated on 4 rows and 2 columns in each cell.
- The ROI window will always include the aimer.

4: Kiosk/presentation application.

- Ignore aimer position, no weight activity.
- Activity calculated on the row and the column in the middle of each cell.
- The ROI window may not include the aimer.
- This is the mode recommended for hand held scanner when used in Presentation mode.

NB : only DPM ROI mode could be used when DPM feature is active.

**See also**

## **25.29. DEC\_SECURITY\_LEVEL, 0x1a002002**

### **Description**

This property controls the reading tolerance of the decoder.

The property value should be set as follows:

0: Very high reading tolerance: this is the most permissive mode. When enabled, the scanner reads codes of variable quality.

1: High reading tolerance

2: Medium reading tolerance: this mode allows medium permissiveness (recommended)

3: Low reading tolerance: this is the least permissive mode

**See also**

### **25.30. DEC\_SUBPIXEL\_FINDER, 0x40100004**

#### **Description**

This property enables the decoder to find bar code symbols that have a very small element size in pixels. The small element size may be due to the printing of a symbol or due to it being imaged from far away. Enabling this property may increase the average time to decode.

This property only affects the following symbologies:

- \* Code 39
- \* Codablock A
- \* Code 128
- \* UPC-A
- \* UPC-E
- \* EAN-13
- \* EAN-8
- \* PDF417

The property values are:

0: disable

1: enable

**See also**

### **25.31. DEC\_TRIGGERED, 0x1a00500a**

#### **Description**

Flag for trigger pull event.

**See also**

### **25.32. DEC\_USE\_DISTANCE\_MAP, 0x40100002**

#### **Description**

This enables an optimization for MLD decoding where instead of doing a straight linear search, a tree type of search is used. Although the benefit is typically orders of magnitude speed improvement, it does introduce a slight possibility of failing to find the right answer.

The property value is a bit field defined as follows:

b2: Enable Distance Maps for UPC

b4: Enable Distance Maps for Code 128

b5: Enable Distance Maps for Code 39

**See also**

DEC\_USE\_MLD

### **25.33. DEC\_USE\_MLD, 0x1a002001**

**Description**

Maximum Likelihood Decoding algorithms.

0: Disable

1: Enable for code 128, code UPC/EAN and code39

Default: 1

**See also**

### **25.34. DEC\_USE\_VESTA, 0x1a00200c**

**Description**

This property specifies whether Vesta algorithms are enabled during the execution of FLD and ID decoders.

The property value is a bit-field defined as follows:

b0: Enable Vesta decoding in FLD

b1: Enable Vesta decoding in ID

Initial value: 0 (Vesta disable)

**See also**

### **25.35. DEC\_VIDEO\_REVERSE\_ENABLED, 0x1a001004**

**Description**

This property specifies whether normal or inverse decoding for linear symbologies is enabled during the execution of Decode. It is recommended to use DEC\_LINEAR\_DECODE\_POLARITY rather than this property.

0: Decode only normal video for 1D codes

1: Decode only inverse video for 1D codes

2: Decode both, normal and inverse video for 1D codes

**Example**

```
/* enable normal video for 1D decoding */
```

```
DecodeSet( DEC_VIDEO_REVERSE_ENABLED, (void *) 0 );
```

**See also**

DEC\_LINEAR\_DECODE\_POLARITY

### 25.36. SD\_PROP\_DPM\_ILLUM\_CORRECTION\_ISEF\_COEFF, 0x40012904

#### Description

control the illumination correction in DPM mode 2 (reflective). This is a temporary property that we expect to deprecate at a later date. In this release, it can be supplied directly to the DCL DecodeSet API function.

**See also**

### 25.37. SD\_PROP\_MISC\_RESERVED\_FLAGS, 0x40005028

#### Description

Add an internal property, SD\_PROP\_MISC\_RESERVED\_FLAGS, to control experimental behavior of the decoder. The functionality of this property is not guaranteed between versions.

In this version:

- Bit 0 == 1 will increase the exclusion radius for Data Matrix when SD\_PROP\_DM\_DOTTED is enabled.
- Bit 1 == 1 implements an alternate search strategy for locating DotCode symbols.

#### Example

```
SD_Set(decoder, SD_PROP_MISC_RESERVED_FLAGS, void *Value);
```

**See also**

## 26. Coupon code properties

### 26.1. DEC\_COUPON\_CODE\_MODE, 0x1a006001

**Description**

Coupon Code decoding mode.

**See also**

## 27. PDF 417 properties

### 27.1. DEC\_PDF417\_ENABLED, 0x1a024001

#### Description

This property specifies whether PDF417 decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable PDF417 decoding

1: Enable PDF417 decoding

Initial Value: 0

#### Example

```
/* enable PDF417 decoding */
```

```
DecodeSet( DEC_PDF417_ENABLED, (void *) 0x01);
```

**See also**

### 27.2. DEC\_PDF417\_MAX\_LENGTH, 0x1a024003

#### Description

Maximum data length to issue a result for PDF417.

**See also**

### 27.3. DEC\_PDF417\_MIN\_LENGTH, 0x1a024002

#### Description

Minimum data length to issue a result for PDF417.

**See also**



## 28. GS1 Composite properties

### 28.1. DEC\_ADD\_SEARCH\_TIME\_UPC\_COMPOSITE, 0x1a003006

**Description**

Maximum additional time in milliseconds to decode the composite piece after reading a UPC.

If null, any UPC label will be decoded without the composite part, whatever the 2D is present or not.

Default: 300ms.

**See also**

DEC\_COMPOSITE\_WITH\_UPC\_ENABLED

### 28.2. DEC\_COMPOSITE\_ENABLED, 0x1a026001

**Description**

Composite code enable.

**See also**

### 28.3. DEC\_COMPOSITE\_WITH\_UPC\_ENABLED, 0x1a026004

**Description**

Allows UPC codes to be read with PDF417 or MicroPDF417 composite.

0: UPC/EAN codes read as UPC/EAN

1: UPC/EAN codes read as GS1 composites

**See also**

DEC\_ADD\_SEARCH\_TIME\_UPC\_COMPOSITE

## 29. Maxicode properties

### 29.1. DEC\_MAXICODE\_ENABLED, 0x1a028001

**Description**

Maxicode enable.

**See also**

### 29.2. DEC\_MAXICODE\_MAX\_LENGTH, 0x1a028003

**Description**

Maximum data length to issue a result for Maxicode.

**See also**

### 29.3. DEC\_MAXICODE\_MIN\_LENGTH, 0x1a028002

**Description**

Minimum data length to issue a result for Maxicode.

**See also**

### 29.4. DEC\_MAXICODE\_SYMBOL\_SIZE, 0x40010602

**Description**

Controls algorithms that aid in decoding of symbols of different sizes.

Initial value: 1

**See also**

### 29.5. DEC\_SD\_PROP\_MC\_MESSAGE\_FORMAT, 0x40010603

**Description**

This property specifies whether the secondary message of a Maxicode is decoded:

0: Primary Message Only

1: Primary + Secondary (if avail)

2: Primary + Secondary Required

Initial value: 1

**See also**

## 30. Datamatrix properties

### 30.1. DEC\_DATAMATRIX\_DOTTED, 0x40010421

#### Description

This property helps on Dotted DataMatrix decodability, especially the ones printed on cigarette packages.

The property value should be set as follows:

0: Disable.

1: Enable.

Initial Value: 0

#### See also

### 30.2. DEC\_DATAMATRIX\_ENABLED, 0x1a029001

#### Description

This property specifies whether Data Matrix decoding is enabled during the execution of the Decoder.

Decoding may be separately enabled or disabled for normal and inverse video symbols. A normal video symbol has a black "L" finder pattern. An inverse video symbol has a white "L" finder pattern. Note that rectangular Data Matrix symbol decoding is separately controlled by the DEC\_DATAMATRIX\_RECTANGLE property.

The property value is a bit field defined as follows:

b0: Enable normal video Data Matrix decoding

b1: Enable inverse video Data Matrix decoding

Note that only the ECC 200 style Data Matrix symbols are decoded by the decoder. The other styles are obsolete.

#### Example

```
/* enable normal and inverse video decoding for datamatrix codes*/  
DecodeSet( DEC_DATAMATRIX_ENABLED, (void *) 3 );
```

#### See also

DEC\_DATAMATRIX\_RECTANGLE

### 30.3. DEC\_DATAMATRIX\_LOW\_CONTRAST, 0x40010414

#### Description

Controls algorithms that improve decoding of low contrast symbols.

Initial value: 1

#### See also

### **30.4. DEC\_DATAMATRIX\_MAX\_LENGTH, 0x1a029003**

#### **Description**

Maximum data length to issue a result for Data Matrix.

**See also**

### **30.5. DEC\_DATAMATRIX\_MIN\_LENGTH, 0x1a029002**

#### **Description**

Minimum data length to issue a result for Data Matrix.

**See also**

### **30.6. DEC\_DATAMATRIX\_NON\_SQUARE\_MODULES, 0x40010412**

#### **Description**

Controls algorithms that improve decoding when individual modules are sufficiently non-square.

**See also**

### **30.7. DEC\_DATAMATRIX\_SHIFTED\_TILES, 0x40010413**

#### **Description**

Controls algorithms that improve decoding when multi-tile symbols are not aligned.

**See also**

### **30.8. DEC\_DATAMATRIX\_SYMBOL\_SIZE, 0x40010416**

#### **Description**

Controls algorithms that improve in decoding of symbols of different sizes.

The property value should be set as follows:

0: Normal Data Matrix operation.

1: Handle smaller Data Matrix symbols.

2: Handle very small Data Matrix symbols.

Initial Value: 1

**See also**

### **30.9. DEC\_DATAMATRIX\_THREADING, 0x40010422**

#### **Description**

This is a Boolean property that enables the use of additional threads within the decoder to prioritize decoding of good quality Data Matrix symbols over those that are damaged and less-likely to decode. If images of Data Matrix symbols commonly contain background elements that loosely resemble Data Matrix, enabling this property can accelerate decoding the true Data Matrix symbol(s).

The property may have a small negative impact on the decoding time of non-Data Matrix symbologies.

This property only has an effect if global multithreading is enabled via DEC\_ID\_MULTITHREADING.

The property value should be set as follows:

0: Disable.

1: Enable.

Initial Value: 0

**See also**

DEC\_ID\_MULTITHREADING

## 31. QR code properties

### 31.1. DEC\_QR\_ENABLED, 0x1a02a001

#### Description

This property specifies whether QR Code decoding is enabled during the execution of Decode.

Decoding may be separately enabled or disabled for normal and inverse video symbols, and for QR Code and Micro QR Code. A normal video symbol is printed in black on a white substrate.

An inverse video symbol is printed in white on a black substrate.

The property value is a bit field defined as follows:

b0: Enable normal video QR Code decoding

b1: Enable inverse video QR Code decoding

b2: Enable normal video Micro QR Code decoding

b3: Enable inverse video Micro QR Code decoding

Initial Value: 0

#### Example

```
/* enable inverse video QR Code and normal video Micro QR Code decoding */  
DecodeSet( DEC_QR_ENABLED, (void *) 0x06);
```

**See also**

### 31.2. DEC\_QR\_MAX\_LENGTH, 0x1a02a003

#### Description

Maximum data length to issue a result for QR Code.

**See also**

### 31.3. DEC\_QR\_MIN\_LENGTH, 0x1a02a002

#### Description

Minimum data length to issue a result for QR Code.

**See also**

### 31.4. DEC\_QR\_NON\_SQUARE\_MODULES, 0x40010902

#### Description

Controls algorithms that improve decoding of QR symbols with non-square modules.

**See also**

### **31.5. DEC\_QR\_SYMBOL\_SIZE, 0x40010904**

**Description**

Controls algorithms that aid in decoding of symbols of different sizes.

Initial value: 1

**See also**

### **31.6. DEC\_QR\_USE\_ALT\_MICROQR\_ID, 0x1b02a003**

**Description**

Specifies whether a Micro-QR symbol will have its own HHP Code ID.

**See also**

### **31.7. DEC\_QR\_WITHOUT\_QZ, 0x40010905**

**Description**

This property allows to decode QR barcodes without quiet zones.

Enabling this property is discouraged by Honeywell, unless absolutely necessary.

The property value should be set as follows:

0: Disable

1: Enable

Initial Value: 0

**Example**

```
/* Allow to decode QR barcodes without quiet zones */
```

```
DecodeSet( DEC_QR_WITHOUT_QZ, (void *)DEC_CONST_ENABLED );
```

**See also**



## 32. Label Code properties

### 32.1. DEC\_LABELCODE\_ENABLED, 0x1a101001

**Description**

Label Code enable.

**See also**

## 33. Postal properties

### Description

Australian Postal Code:

- 4-State barcode, can be 37, 52 or 67 bars (11, 16 or 21 codewords)
- start and stop are 2 bars (tracker ascender + tracker)
- The 4 bars after the start are the Format Control Code (FCC)  
FCC supported by the decoder:
  - 37 bars: FCC01..FCC32, FCC34..FCC37, FCC45, FCC87, FCC92
  - 52 bars: FCC00, FCC33, FCC38..FCC43, FCC46..FCC61, FCC72
  - 67 bars: FCC00, FCC44, FCC62..FCC71, FCC73..FCC91, FCC93..FCC99
- Customer information field (16 or 31 bars)
- Reed Solomon error correction (12 bars)

POSTNET (POSTal Numeric Encoding Technique)

- 2-State barcode, can be
  - 32 bars: 5-digit ZIP code (A Field)
  - 37 bars: 6-digit, (B Field), now obsolete
  - 47 bars: 8-digit, (B' Field)
  - 52 bars: 9-digit ZIP+4 code (C Field)
  - 62 bars: 11-digit, ZIP Code, ZIP+4 Code, Delivery Point Code (C' Field)
- start and stop are just one high bar (also named Frame bars)
- each digit is encoded with 5 bars, 2 high and 3 low (same as CEPNET)
- correction character: Modulo 10, 5 bars

CEPNET:

- 2-State barcode, can be 47, 72 bars (8 or 13 digits)
- start and stop are just one high bar (also named Delimiter)
- each digit is encoded with 5 bars, 2 high and 3 low (same as POSTNET)
- correction character: Modulo 10, 5 bars

PLANET (Postal Alpha Numeric Encoding Technique)

- fully superseded by Intelligent Mail Barcode by January 28, 2013.
- 2-State barcode, can be 62 or 72 bars (12- or 14-digit including checksum)
  - Service Type: 2 digits
  - Customer ID: 9 or 11 digits
  - Checksum: 1 digit
- start and stop are just one high bar (also named Delimiter)
- each digit is encoded with 5 bars, 3 high and 2 low (exact opposite of POSTNET)
- correction character: Modulo 10, 5 bars

Canada Post (CPC 4-State or PostBar)

- derived from the RM4SCC
- 3 types:
  - 52 bars (Business reply): 12 reed-solomon parity check bars, 36 information bars (12 characters)
  - 56 bars: 30 reed-solomon parity check bars, 18 information bars (6 characters)
  - 73 bars (not supported by ID)
  - 82 bars: 12 reed-solomon parity check bars, 66 information bars (22 characters)
- 4 characters set can be used
- start and stop are 2 bars (tracker ascender + tracker)

RM4SCC (Royal Mail 4-State Customer Code), also known as CBC (Customer Code)

- start is tracker ascender, stop is full tracker
- checksum based on modulo 6
- 9 characters max encoded (38 valid characters: numeric, alphanumeric upper case and () or [])
  - each character consists of 4 bars, Two of these have ascenders and two have descenders.
- 3 types known:
  - 51 bars: RED TAG Barcode
  - 66 bars: Mailmark barcodes C (consolidated)
  - 78 bars: Mailmark barcodes L (long)

KIX (Klant index or Customer index)

- 4-State
- Slightly modified from CBC (doesn't use the start and end symbols or the checksum, separates the house number and suffixes with an X)

Intelligent Mail Barcode (IM barcode), also known as 4-State Customer Barcode (4CB or 4-CB or USPS4CB)

- 4-State
- 65-bar barcode for use on mail in the United States
- Supersede POSTNET and PLANET.

InfoMail Barcode A

- 4-State, 51 bars (17 cw, 12 data + 5 error correction)
- Reed-solomon error correction

UPU ID-tag 4-state (S18d)

- 4-state
- can be 57 or 75 bars

POSTI 4-state, Finnish post barcode

- 4-state, 42 bars (16 cw, 8 data + 8 error correction)
- Reed-solomon error correction

### **33.1. DEC\_AUS\_POST\_BAR\_OUTPUT\_ENABLED, 0x40010817**

#### **Description**

Issues the bar number instead of decoded data for Australia Post.

**See also**

### **33.2. DEC\_AUS\_POST\_INTERPRET\_MODE , 0x40010818**

#### **Description**

Australia Post Customer data interpretation mode.

**See also**

### **33.3. DEC\_AUS\_POST\_ZERO\_FCC, 0x40010813**

**Description**

Allows decoding of Australia Post code with no FCC information.

**See also**

### **33.4. DEC\_CAN\_POST\_BAR\_OUTPUT, 0x40010819**

**Description**

Issues the bar number instead of the decoded data for Canadian Post.

**See also**

### **33.5. DEC\_GOCR\_PHONENUMBER, 0x40012314**

**Description**

Enable Recognition of US phone numbers.

**See also**

### **33.6. DEC\_GOCR\_ZIPCODE, 0x40012313**

**Description**

Enable Recognition of US zip codes.

**See also**

### **33.7. DEC\_PLANETCODE\_CHECK\_DIGIT\_TRANSMIT, 0x1a130001**

**Description**

Planetcode check digit transmit enable.

**See also**

### **33.8. DEC\_POSTAL\_ENABLED, 0x1a110001**

## Description

This property sets which, if any, postal decoding is enabled during the execution of Decode.

The property value should be set as follows:

- 0: Disabled
- 1: Australia Post (Australia 4-state)
- 2: Royal Mail (InfoMail)
- 3: Japan Post (Japan 4-state)
- 4: Dutch Post (KIX code)
- 5: US Postal (Planet)
- 6: US Postal (Postnet)
- 7: Royal Mail (Customer Code)
- 8: Royal Mail (InfoMail, Customer Code)
- 9: US Postal (UPU)
- 10: US Postal (Intelligent Mail)
- 11: US Postal (Postnet B Fields)
- 12: US Postal (Planet, Postnet)
- 13: US Postal (Planet, UPU)
- 14: US Postal (Postnet, UPU)
- 15: US Postal (Planet, Intelligent Mail)
- 16: US Postal (Postnet, Intelligent Mail)
- 17: US Postal (UPU, Intelligent Mail)
- 18: US Postal (Planet, Postnet B Fields)
- 19: US Postal (Postnet B Fields, UPU)
- 20: US Postal (Postnet B Fields, Intelligent Mail)
- 21: US Postal (Planet, Postnet, UPU)
- 22: US Postal (Planet, Postnet, Intelligent Mail)
- 23: US Postal (Planet, UPU, Intelligent Mail)
- 24: US Postal (Postnet, UPU, Intelligent Mail)
- 25: US Postal (Planet, Postnet B Fields, UPU)
- 26: US Postal (Planet, Postnet B Fields, Intelligent Mail)
- 27: US Postal (Postnet, Postnet B Fields, UPU, Intelligent Mail)
- 28: US Postal (Planet, Postnet, UPU, Intelligent Mail)
- 29: US Postal (Planet, Postnet B Fields, UPU, Intelligent Mail)
- 30: Canada Post (Canada 4-state)
- 31: Brazil Post (CEPNet)
- 32: Sweden Post (Sweden 4-state)
- 33: Brazil Post (Brazil 4-state)
- 34: Brazil Post (Brazil 4-state, CEPNet)
- 35: Royal Mail (Mailmark)
- 36: Royal Mail (Mailmark, InfoMail)
- 37: Royal Mail (Mailmark, Customer Code)
- 38: Royal Mail (Mailmark, InfoMail, Customer Code)
- 39: Finland Post (Posti)
- 40: Portugal Post (Portugal 4-state)
- 41: Portugal Post (Portugal 4-state, UPU)
- 42: New Zealand Post (New Zealand 4-state)

Initial Value: 0

## Example

```
/* enable Postnet and UPU decoding */  
DecodeSet( DEC_POSTAL_ENABLED, (void *)14);
```

**See also**

DEC\_POSTAL\_ENABLED\_DIRECT

### **33.9. DEC\_POSTNET\_CHECK\_DIGIT\_TRANSMIT, 0x1a120001**

**Description**

Postnet check digit transmit enable.

**See also**

### **33.10. DEC\_ROYAL\_MAIL\_FORMAT\_CHECK\_MIN\_LENGTH, 0x40010815**

**Description**

Verifies the minimum length of Royal Mail.

**See also**

## 34. Grid Matrix properties

### 34.1. DEC\_GRIDMATRIX\_ENABLED, 0x1a160001

#### Description

This property specifies whether Grid Matrix decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Grid Matrix decoding

1: Enable Grid Matrix decoding

Initial Value: 0

#### Example

```
/* enable Grid Matrix decoding */
```

```
DecodeSet( DEC_GRIDMATRIX_ENABLED, (void *) DEC_CONST_ENABLED);
```

#### See also

### 34.2. DEC\_GRIDMATRIX\_MAX\_LENGTH, 0x1a160003

#### Description

Sets the maximum data length of a Grid Matrix that is allowed to issue.

#### Example

```
/* set Grid Matrix max length to 500 */
```

```
DecodeSet( DEC_GRIDMATRIX_MAX_LENGTH, (void *) 500);
```

#### See also

DEC\_GRIDMATRIX\_MIN\_LENGTH

### 34.3. DEC\_GRIDMATRIX\_MIN\_LENGTH, 0x1a160002

#### Description

Sets the minimum data length of a Grid Matrix that is allowed to issue.

#### Example

```
/* set Grid Matrix min length to 7 */
```

```
DecodeSet( DEC_GRIDMATRIX_MIN_LENGTH, (void *) 7);
```

#### See also

DEC\_GRIDMATRIX\_MAX\_LENGTH

## 35. Linear properties

### 35.1. DEC\_LINEAR\_DAMAGE\_IMPROVEMENTS, 0x40005025

**Description**

enable/disable of enhanced methods for dealing with heavily damaged linear bar codes (defaults to 0)

**See also**

### 35.2. DEC\_LINEAR\_EXTENSIVE\_SEARCH, 0x40100016

**Description**

This property expands the region around the search center in which the decoder will look for linear barcodes. This can be particularly useful when trying to read multiple barcodes that are relatively close to the search center. This property applies only if DEC\_CYCLING\_FINDER is enabled and DEC\_PASS\_THROUGH is disabled.

The property values are:

0: disable

1: enable

Initial Value: 0

**See also**

DEC\_CYCLING\_FINDER

DEC\_PASS\_THROUGH

### 35.3. DEC\_LINEAR\_FINDER\_TYPE, 0x40100014

**Description**

This property controls the type of finder the decoder uses to locate linear symbols.

The property values are:

0: The decoder attempts to locate linear symbols using a set of fixed thresholds.

1: The decoder attempts to locate linear symbols using a derivative-based approach.

Initial value: 0

**See also**



## 36. DotCode properties

### Description

DotCode parameters.

### 36.1. DEC\_DOTCODE\_ENABLED, 0x1a161001

#### Description

This property specifies whether DotCode decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable DotCode decoding

1: Enable DotCode decoding

Initial Value: 0

#### Example

```
/* enable DotCode decoding */
```

```
DecodeSet( DEC_DOTCODE_ENABLED, (void *) DEC_CONST_ENABLED);
```

**See also**

### 36.2. DEC\_DOTCODE\_EXTENSIVE\_SEARCH, 0x40016004

#### Description

This property allows a more extensive search of a DotCode barcode. It helps to decode bad printed barcode but increases DotCode CPU usage.

The property value should be set as follows:

0: Disable DotCode extensive search

1: Enable DotCode extensive search

Initial Value: 0

#### Example

```
/* enable DotCode extensive search */
```

```
DecodeSet( DEC_DOTCODE_EXTENSIVE_SEARCH, (void *) DEC_CONST_ENABLED);
```

**See also**

### 36.3. DEC\_DOTCODE\_MAX\_LENGTH, 0x1a161003

#### Description

Sets the maximum data length of a DotCode that is allowed to issue.

#### Example

```
/* set DotCode max length to 500 */  
DecodeSet( DEC_DOTCODE_MAX_LENGTH, (void *) 500);
```

**See also**

DEC\_DOTCODE\_MIN\_LENGTH

### **36.4. DEC\_DOTCODE\_MIN\_LENGTH, 0x1a161002**

**Description**

Sets the minimum data length of a DotCode that is allowed to issue.

**Example**

```
/* set DotCode min length to 7 */  
DecodeSet( DEC_DOTCODE_MIN_LENGTH, (void *) 7);
```

**See also**

DEC\_DOTCODE\_MAX\_LENGTH

### **36.5. DEC\_DOTCODE\_SEARCH\_AREA, 0x40016008**

**Description**

This property allows to select the size of the search area for Dotcode:

- 0: (small) a centered window of 270x200 pixels (max)
- 1: (medium) a centered window of 430x270 pixels (max)
- 3: (large) a centered window of 60% of the input image
- 4: (full) a centered window of 90% of the input image

Initial Value: 0

If a ROI mode is used, the search area is applied on the ROI.

The decode duration increases with the search area. It is recommended to keep the search area as small as possible (ROI mode, especially mode 2, can also be used for this).

**See also**

DEC\_DOTCODE\_ENABLED

DEC\_DOTCODE\_EXTENSIVE\_SEARCH

DEC\_ROI\_MODE

## 37. Monocolor properties

### 37.1. DEC\_MONOCOLOR\_ENABLED, 0x1a007001

#### Description

Monocolor interpolation enable.

#### See also

## 38. Plessey properties

### Description

UK Plessey

### 38.1. DEC\_PLESSEY\_ENABLED, 0x1a162001

#### Description

This property specifies whether UK Plessey decoding is enabled during the execution of Decode.

The property value should be set as follows:

0: Disable Plessey decoding

1: Enable Plessey decoding

Initial Value: 0

**See also**

### 38.2. DEC\_PLESSEY\_MAX\_LENGTH, 0xfd013002

#### Description

Maximum data length to issue a result for UK Plessey.

Range: 0..80

Initial value: 80

**See also**

### 38.3. DEC\_PLESSEY\_MIN\_LENGTH, 0xfd013003

#### Description

Minimum data length to issue a result for UK Plessey.

Range: 0..80

Initial value: 4

**See also**

## 39. Not assigned yet properties

### 39.1. DEC\_ADDENDA\_INCREASED\_SEPARATION, 0x0A010009

#### Description

Allow big separation between main code and addenda.

0: Disable.

1: Enable.

Property Data Type: int

Set By: Value

Initial Value: 0

#### Example

```
SD_Set(Handle, DEC_ADDENDA_INCREASED_SEPARATION, (void *) SD_CONST_ENABLED);
```

#### See also