Exhibit L: User Manual - Part 2 of 2

FCC ID: HN22011B-2

Code 128 Enumerations

```
typedef enum tagCode128Decoding
ITC CODE128 NOTACTIVE = 0, // Default
ITC CODE128 ACTIVE = 1,
ITC CODE128 NO CHANGE = 255
} ITC CODE128 DECODING;
typedef enum tagEan128Identifier
ITC EAN128 ID REMOVE,
ITC_EAN128_ID_INCLUDE, // Default
ITC_EAN128_ID_NO_CHANGE = 255
} ITC_EAN128_IDENTIFIER;
typedef enum tagCode128Cip128
{
ITC CODE128 CIP128 NOTACTIVE = 0, // Default
ITC CODE128 CIP128 ACTIVE = 1,
ITC_CODE128_CIP128_NO_CHANGE = 255
} ITC CODE128 CIP128;
```

#define ITC_CODE128_FNC1_NO_CHANGE 255.
This definition can be used when the Code128 FNC1 does not require any change.

 $\# define \ \mbox{ITC} \ \mbox{BC} \ \mbox{LENGTH} \ \mbox{NO} \ \mbox{CHANGE} \ 255.$ This definition can be used when the bar code length does not require any change.

The table below shows what to be expected for EAN 128 labels for various symbology identifier transmit configurations and EAN 128 Identifier options.

Setup			Application's Expected Result		
EAN 12	8]C1 ID	Symbology ID option	EAN 128 Label	Other Labels	
1	Include]C1	Disabled	<data></data>	<data></data>	
2	Remove]C1	Disabled	<data></data>	<data></data>	
3	Include]C1	AIM ID Transmitted]C1 <data></data>]XY <data></data>	
4	Remove]C1	AID ID Transmitted]C1 <data></data>]XY <data></data>	
5	Include]C1	Custom ID Transmitted	Z]C1 <data></data>	Z <data></data>	
6	Remove]C1	Custom ID Transmitted	Z <data></data>	Z <data></data>	
where "X" is the symbology identifier, "Y" is the modifier character, and "Z" is the 1-byte symbology identifier.					

IS9CConfig::Getl2of5

This function retrieves the current settings of Interleaved 2 of 5.

Syntax

HRESULT IS9CConfig::GetI2of5(ITC_INTERLEAVED2OF5_DECODING* peDecode, ITC_INTERLEAVED2OF5_CHECK_DIGIT* peCheck, ITC_BARCODE_LENGTH_ID* peLengthId, BYTE rbgLengthBuff[], DWORD* pdwNumBytes);

Parameters

peDecode	[out]	Pointer to the ITC_INTERLEAVED2OF5_DECODING location to receive the decoding for Interleaved 2 of 5 symbology.
peCheck	[out]	Pointer to the ITC_INTERLEAVED2OF5_CHECK_DIGIT location to receive the check digit.
peLengthId	[out]	Pointer to the ITC_BARCODE_LENGTH_ID location to receive an indicator of either ITC_BARCODE_LENGTH or ITC_BARCODE_FIXED_LENGTH.
rgbLengthBuff	[out,size	e_is(3)] An array of bytes to receives 1 byte of data for ITC_BARCODE_LENGTH or 3 bytes of data for ITC_BARCODE_FIXED_LENGTH.
pdwNumBytes	[out]	Pointer to the DWORD location to receive a number indicating number of bytes in <i>rbgLengthBuff[]</i> : 1 byte for ITC_BARCODE_LENGTH or 3 bytes for ITC_BARCODE_FIXED_LENGTH.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

IS9CConfig::SetI2of5

This function updates the Interleaved 2 of 5 settings with new values.

Syntax

HRESULT IS9CConfig::SetI2of5(ITC_INTERLEAVED2OF5_DECODING eDecode, ITC_INTERLEAVED2OF5_CHECK_DIGIT eCheck, ITC_BARCODE_LENGTH_ID_eLengthId, BYTE_rgbLengthBuff[], DWORD dwNumBytes);

Parameters

eDecode	[in]	Identifies the decoding for Interleaved 2 of 5 symbology.
eCheck	[in]	Identifies the check digit.
eLengthId	[in]	Use ITC_BARCODE_LENGTH_NO_CHANGE to indicate no change for bar code length. Use ITC_BARCODE_LENGTH for any length and minimum length, and set <i>rgbLengthBuff[0]</i> to a valid length value. Use ITC_BARCODE_FIXED_LENGTH to compose 1 or 2 or 3 fixed lengths, and set 3 bytes: <i>rgbLengthBuff[0]</i> , <i>rgbLengthBuff[1]</i> , <i>rgbLengthBuff[2]</i> with valid values.
rgbLengthBuff	[in,si	ze_is(dwNumBytes)] Contains bar code lengths when <i>eLengthId</i> = Use ITC_BARCODE_LENGTH or Use ITC_BARCODE_FIXED_LENGTH.
dwNumBytes	[in]	Number of bytes in <i>rbgLengthBuff</i>]. For S9C, this value is 1 when <i>eLengthId</i> = ITC_BARCODE_LENGTH or 3 when <i>eLengthId</i> = ITC_BARCODE_FIXED_LENGTH.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

Interleaved 2 of 5 Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_INTERLEAVED2OF5_DECODING
Check Digit	Not Used	ITC_INTERLEAVED2OF5_CHECK_DIGIT
Bar Code Length	Minimum Length = 6	0x00`0xFE ITC_BC_LENGTH_NO_CHANGE

Interleaved 2 of 5 Enumerations

```
typedef enum tagInterleaved2of5Decoding
                                   // Default
ITC INTERLEAVED2OF5 NOTACTIVE = 0,
ITC INTERLEAVED2OF5 ACTIVE = 1,
ITC INTERLEAVED20F5 NO CHANGE = 255
} ITC INTERLEAVED20F5 DECODING;
typedef enum tagInterleaved2of5CheckDigit
ITC INTERLEAVED2OF5 CHECK NOTUSED,
                                       // Default
ITC_INTERLEAVED2OF5_CHECK_MOD10_XMIT,
ITC_INTERLEAVED2OF5_CHECK_MOD10_NOTXMIT,
ITC_INTERLEAVED2OF5_CHECK_FRENCH_CIP_XMIT,
ITC INTERLEAVED20F5 CHECK FRENCH CIP NOTXMIT,
ITC INTERLEAVED2OF5 CHECK NO CHANGE = 255
} ITC INTERLEAVED20F5 CHECK DIGIT;
typedef enum tagBarcodeLengthId
ITC BARCODE LENGTH = 0,
ITC BARCODE FIXED LENGTH,
ITC BARCODE LENGTH NO CHANGE = 255
} ITC BARCODE LENGTH ID;
```

IS9CConfig::GetMatrix2of5

This function retrieves the current settings of Matrix 2 of 5.

Syntax

HRESULT IS9CConfig::GetMatrix2of5(ITC_MATRIX2OF5_DECODING*
 peDecode, DWORD* pdwLength);

Parameters

peDecode	[out]	Pointer to the ITC_MATRIX2OF5_DECODING location to receive the decoding for Matrix 2 of 5 symbology.
pdwLength	[out]	Pointer to the DWORD location to receive a value for the bar code length.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

IS9CConfig::SetMatrix2of5

This function updates the Matrix 2 of 5 settings with new values.

Syntax

HRESULT IS9CConfig::SetMatrix2of5(ITC_MATRIX2OF5_DECODING
eDecode, DWORD dwLength);

Parameters

eDecode [in] Identifies the decoding for Matrix 2 of 5 symbology.

dwLength [in] Identifies the bar code length.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

Matrix 2 of 5 Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_MATRIX2OF5_DECODING
Bar Code Length	Minimum Length = 6	0x00`0xFE ITC_BC_LENGTH_NO_CHANGE

Matrix 2 of 5 Enumerations

typedef enum tagMatrix2of5Decoding
{
 ITC_MATRIX2OF5_NOTACTIVE = 0, // Default
 ITC_MATRIX2OF5_ACTIVE = 1,
 ITC_MATRIX2OF5_NO_CHANGE = 255
 ITC_MATRIX2OF5_DECODING;
 #define ITC_BC_LENGTH_NO_CHANGE 255. This definition can be used when the bar
 code length does not require any change.

IS9CConfig::GetMSI

This function retrieves the current MSI settings.

Syntax

HRESULT IS9CConfig::GetMSI(ITC_MSI_DECODING* peDecode, ITC MSI CHECK DIGIT* peCheck, DWORD* pdwLength);

Parameters

peDecode	[out]	Pointer to the ITC_MSI_DECODING location to receive the decoding for MSI symbology.
peCheck	[out]	Pointer to the ITC_MSI_CHECK_DIGIT location to receive the check digit.
pdwLength	[out]	Pointer to the DWORD location to receive the bar code length.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

IS9CConfig::SetMSI

This function updates the MSI settings with new values.

Syntax

HRESULT **IS9CConfig::SetMSI(** ITC_MSI_DECODING *eDecode*, ITC_MSI_CHECK_DIGIT *eCheck*, DWORD *dwLength*);

Parameters

eDecode	[in]	Identifies the decoding for MSI symbology.
eCheck	[in]	Identifies the check digit.
dwLength	[in]	Identifies the bar code length.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

MSI Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_MSI_DECODING
Check Digit	MOD 10 checked and transmitted	ITC_MSI_CHECK_DIGIT
Bar Code Length	Minimum Length = 6	0x00`0xFE ITC_BC_LENGTH_NO_CHANGE

MSI Enumerations

```
typedef enum tagMsiDecoding
{
ITC_MSI_NOTACTIVE = 0, // Default
ITC_MSI_ACTIVE = 1,
ITC_MSI_OCHANGE = 255
} ITC_MSI_DECODING;
typedef enum tagMsiCheckDigit
{
ITC_MSI_CHECK_MOD10_XMIT, // Default
ITC_MSI_CHECK_MOD10_NOTXMIT,
ITC_MSI_CHECK_DOUBLEMOD10_XMIT,
ITC_MSI_CHECK_DOUBLEMOD10_NOTXMIT,
ITC_MSI_CHECK_DOUBLEMOD10_NOTXMIT,
ITC_MSI_CHECK_DIGIT;
#define ITC_BC_LENGTH_NO_CHANGE 255. This definition can be used when the bar
code length does not require any change.
```

IS9CConfig::GetPDF417

This function retrieves the current PDF417 settings.

Syntax

HRESULT IS9CConfig::GetPDF417 (ITC_PDF417_DECODING* pePdf417Decode, ITC_PDF417_MACRO_PDF* peMacroPdf, ITC_PDF417_CTRL_HEADER* pePdfControlHeader, ITC_PDF417_FILE_NAME* pePdfFileName, ITC_PDF417_SEGMENT_COUNT* pePdfSegmentCount, ITC_PDF417_TIME_STAMP* pePdfTimeStamp, ITC_PDF417_SENDER* pePdfSender, ITC_PDF417_ADDRESSEE* pePdfAddressee, ITC_PDF417_FILE_SIZE* pePdfFileSize, ITC_PDF417_CHECKSUM* pePdfChecksum);

Parameters

pePdf417Decode	[out]	Pointer to the ITC_PDF417_DECODING location to receive the decoding for PDF417 symbology.
peMacroPdf	[out]	Pointer to the ITC_PDF417_MACRO_PDF location to receive the Macro PDF.
pePdfControlHeader	[out]	Pointer to the ITC_PDF417_CTRL_HEADER location to receive the control header.
pePdfFileName	[out]	Pointer to the ITC_PDF417_FILE_NAME location to receive the file name.
pePdfSegmentCount	[out]	Pointer to the ITC_PDF417_SEGMENT_COUNT location to receive the segment count.
pePdfTimeStamp	[out]	Pointer to the ITC_PDF417_TIME_STAMP location to receive the time stamp.

700 Series Color Mobile Computer User's Manual

pePdfSender	[out]	Pointer to the ITC_PDF417_SENDER location to receive the sender.
pePdfAddressee	[out]	Pointer to the ITC_PDF417_ADDRESSEE location to receive the addressee.
pePdfFileSize	[out]	Pointer to the ITC_PDF417_FILE_SIZE location to receive the file size.
pePdfChecksum	[out]	Pointer to the ITC_PDF417_CHECKSUM location to receive the checksum.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

IS9CConfig::SetPDF417

This function updates the PDF417 settings with new values.

Syntax

```
HRESULT IS9CConfig::SetPDF417 ( ITC_PDF417_DECODING

ePdf417Decode, ITC_PDF417_MACRO_PDF_eMacroPdf,

ITC_PDF417_CTRL_HEADER ePdfControlHeader,

ITC_PDF417_FILE_NAME ePdfFileName, ITC_PDF417_SEGMENT_COUNT

ePdfSegmentCount, ITC_PDF417_TIME_STAMP_ePdfTimeStamp,

ITC_PDF417_SENDER ePdfSender, ITC_PDF417_ADDRESSEE

ePdfAddressee, ITC_PDF417_FILE_SIZE_ePdfFileSize,

ITC_PDF417_CHECKSUM_ePdfChecksum);
```

Parameters

ePdf417Decode	[in]	Identifies the decoding for PDF417 symbology.
eMacroPdf	[in]	Identifies the Macro PDF.
ePdfControlHeader	[in]	Identifies the control header.
ePdfFileName	[in]	Identifies the file name.
ePdfSegmentCount	[in]	Identifies the segment count.
ePdfTimeStamp	[in]	Identifies the time stamp.
ePdfSender	[in]	Identifies the sender.
ePdfAddressee	[in]	Identifies the addressee.
ePdfFileSize	[in]	Identifies the file size.
ePdfChecksum	[in]	Identifies the checksum.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also None.

PDF 417 Default Settings

Parameter	Default	Valid Range		
Decoding	Not Active	ITC_PDF417_DECODING		
Macro PDF	Macro PDF Buffered	ITC_PDF417_MACRO_PDF		
Control Header	Not Transmitted	ITC_PDF417_CTRL_HEADER		
*File Name	Not Transmitted	ITC_PDF417_FILE_NAME ITC_PDF417_SEGMENT_COUNT ITC_PDF417_TIME_STAMP ITC_PDF417_SENDER		
*Segment Count	Not Transmitted			
*Time Stamp	Not Transmitted			
*Sender	Not Transmitted			
*Address	Not Transmitted	ITC_PDF417_ADDRESSEE		
*File Size	Not Transmitted	ITC_PDF417_FILE_SIZE		
*Check Sum	Not Transmitted	ITC_PDF417_CHECKSUM		
* These are Macro PDF Optional Fields.				

PDF 417 Enumerations

```
typedef enum tagPdf417Decoding
{
ITC PDF417 NOTACTIVE = 0,
ITC PDF417 ACTIVE = 1,
                                    // Default
ITC PDF417 NO CHANGE = 255
} ITC PDF417 DECODING;
typedef enum tagPdf417MacroPdf
ITC PDF417 MACRO UNBUFFERED = 0,
ITC PDF417 MACRO BUFFERED = 1,
                                    // Default
ITC PDF417 MACRO NO CHANGE = 255
} ITC PDF417 MACRO PDF;
typedef enum tagPdf417ControlHeader
ITC PDF417 CTRL HEADER NOTXMIT = 0,
                                       // Default
ITC PDF417 CTRL HEADER XMIT = 1,
ITC PDF417 CTRL HEADER NO CHANGE = 255
} ITC PDF417 CTRL HEADER;
typedef enum tagPdf417FileName
ITC PDF417 FILE NAME NOTXMIT = 0, // Default
ITC PDF417 FILE NAME XMIT = 1,
ITC PDF417 FILE NAME NO CHANGE = 255
} ITC_PDF417_FILE_NAME;
typedef enum tagPdf417SegmentCount
{
ITC PDF417 SEGMENT COUNT NOTXMIT = 0, // Default
ITC PDF417 SEGMENT COUNT XMIT = 1,
```

```
ITC PDF417 SEGMENT COUNT NO CHANGE = 255
} ITC PDF417 SEGMENT COUNT;
typedef enum tagPdf417TimeStamp
ITC_PDF417_TIME_STAMP_NOTXMIT = 0, // Default
ITC_PDF417_TIME_STAMP_XMIT = 1,
ITC_PDF417_TIME_STAMP_NO_CHANGE = 255
} ITC_PDF417_TIME_STAMP;
typedef enum tagPdf417Sender
{
ITC PDF417 SENDER NOTXMIT = 0, // Default
ITC PDF417 SENDER XMIT = 1,
ITC PDF417 SENDER NO CHANGE = 255
} ITC_PDF417_SENDER;
typedef enum tagPdf417Addressee
                                    // Default
ITC PDF417 ADDRESSEE NOTXMIT = 0,
ITC_PDF417_ADDRESSEE_XMIT = 1,
ITC_PDF417_ADDRESSEE_NO_CHANGE = 255
} ITC_PDF417_ADDRESSEE;
typedef enum tagPdf417FileSize
ITC PDF417 FILE SIZE NOTXMIT = 0, // Default
ITC PDF417 FILE SIZE XMIT = 1,
ITC PDF417 FILE SIZE NO CHANGE = 255
} ITC PDF417 FILE SIZE;
typedef enum tagPdf417Checksum
{
ITC PDF417 CHECKSUM NOTXMIT = 0,
                                   // Default
ITC PDF417 CHECKSUM XMIT = 1,
ITC PDF417 CHECKSUM NO CHANGE = 255
} ITC_PDF417_CHECKSUM;
```

IS9CConfig::GetPlessey

This function retrieves the current Plessey settings.

Syntax

```
HRESULT IS9CConfig::GetPlessey( ITC_PLESSEY_DECODING*
peDecode, ITC_PLESSEY_CHECK_DIGIT* peCheck, DWORD* pdwLength
);
```

Parameters

peDecode	[out]	Pointer to the ITC_PLESSEY_DECODING location to receive the decoding for Plessey symbology.
peCheck	[out]	Pointer to the ITC_PLESSEY_CHECK_DIGIT location to receive the check digit.
pdwLength	[out]	Pointer to the DWORD location to receive the bar code length.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

IS9CConfig::SetPlessey

This function updates the Plessey settings with new values.

Syntax

```
HRESULT IS9CConfig::SetPlessey( ITC_PLESSEY_DECODING
eDecode, ITC PLESSEY CHECK DIGIT eCheck, DWORD dwLength );
```

Parameters

eDecode	[in]	Identifies the decoding for Plessey symbology.
eCheck	[in]	Identifies the check digit.
dwLength	[in]	Identifies the bar code length.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

Plessey Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_PLESSEY_DECODING
Check Digit	Not Transmitted	ITC_PLESSEY_CHECK_DIGIT
Bar Code Length	Any Bar Code Length	0x00`0xFE ITC_BC_LENGTH_NO_CHANGE

Plessey Enumerations

IS9CConfig::GetStandard2of5

This function retrieves the current Standard 2 of 5 settings.

Syntax

```
HRESULT IS9CConfig::GetStandard2of5(
ITC_STANDARD2OF5_DECODING* peDecode,
ITC_STANDARD2OF5_FORMAT* peFormat,
ITC_STANDARD2OF5_CHECK_DIGIT* peCheck,
ITC_BARCODE_LENGTH_ID* peLengthId, BYTE rgbLengthBuff,
DWORD* pdwNumBytes );
```

Parameters

peDecode	[out]	Pointer to the ITC_STANDARD2OF5_DECODING location to receive the decoding for Standard 2 of 5 symbology.
peFormat	[out]	Pointer to the ITC_STANDARD2OF5_FORMAT location to receive the format.
peCheck	[out]	Pointer to the ITC_STANDARD2OF5_CHECK_DIGIT location to receive Modulo 10 check digit.
peLengthId	[out]	Pointer to the ITC_BARCODE_LENGTH_ID location to receive an indicator of either ITC_BARCODE_LENGTH or ITC_BARCODE_FIXED_LENGTH.
rgbLengthBuff	[out,siz	e_is(3)] An array of bytes to receives 1 byte of data for ITC_BARCODE_LENGTH, or 3 bytes of data for ITC_BARCODE_FIXED_LENGTH.
pdwNumBytes	[out]	Pointer to the DWORD location to receive a number indicating number of bytes in <i>rbgLengthBuff[]</i> : 1 byte for ITC_BARCODE_LENGTH or 3 bytes for ITC_BARCODE_FIXED_LENGTH.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

IS9CConfig::SetStandard2of5

This function updates the Standard 2 of 5 settings with new values.

Syntax

HRESULT IS9CConfig::SetStandard2of5(

```
ITC_STANDARD2OF5_DECODING eDecode, ITC_STANDARD2OF5_FORMAT
eFormat, ITC_STANDARD2OF5_CHECK_DIGIT eCheck,
ITC_BARCODE_LENGTH_ID eLengthId, BYTE rgbLengthBuff[], DWORD
dwNumBytes );
```

Parameters

eDecode	[in]	Identifies the decoding for Standard 2 of 5 symbology.
eFormat	[in]	Identifies the format.
eCheck	[in]	Identifies the Modulo 10 check digit.
eLengthId	[in]	Use ITC_BARCODE_LENGTH_NO_CHANGE to indicate no change for bar code length. Use ITC_BARCODE_LENGTH for any length and minimum length, and set rgbLengthBuff[0] to a valid length value. Use ITC_BARCODE_FIXED_LENGTH to compose 1 or 2 or 3 fixed lengths, and set 3 bytes: rgbLengthBuff[0], rgbLengthBuff[1], rgbLengthBuff[2] with valid values.
rgbLengthBuff	[in,s	ize_is(dwNumBytes)] An array of bytes containing bar code lengths when <i>eLengthId</i> = ITC_BARCODE_LENGTH or ITC_BARCODE_FIXED_LENGTH.
dwNumBytes	[in]	Number of bytes in <i>rbgLengthBuff</i>]. For S9C, this value is 1 when <i>eLengthId</i> = ITC_BARCODE_LENGTH or 3 when <i>eLengthId</i> = ITC_BARCODE_FIXED_LENGTH.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

Standard 2 of 5 Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_STANDARD2OF5_DECODING
Format	Identicon (6 Start/Stop bars)	ITC_STANDARD2OF5_FORMAT
Check Digit	Not Used	ITC_STANDARD2OF5_CHECK_DIGIT
Bar Code Length	Minimum Length = 6	0x00-0xFE ITC_BC_LENGTH_NO_CHANGE

Standard 2 of 5 Enumerations

```
typedef enum tagStandard2of5Decoding
£
ITC STANDARD2OF5 NOTACTIVE = 0, // Default
ITC STANDARD2OF5 ACTIVE = 1,
ITC STANDARD20F5 NO CHANGE = 255
} ITC STANDARD20F5 DECODING;
typedef enum tagStandard2of5Format
{
ITC STANDARD2OF5 FORMAT IDENTICON, // Default
ITC_STANDARD2OF5_FORMAT_COMPUTER_IDENTICS,
ITC STANDARD2OF5 FORMAT NO CHANGE = 255
} ITC STANDARD20F5 FORMAT;
typedef enum tagStandard2of5CheckDigit
{
ITC STANDARD2OF5 CHECK NOTUSED, // Default
ITC STANDARD20F5 CHECK XMIT,
ITC STANDARD20F5 CHECK NOTXMIT,
ITC STANDARD20F5 CHECK NO CHANGE = 255
} ITC STANDARD2OF5 CHECK DIGIT;
typedef enum tagBarcodeLengthId
{
ITC BARCODE LENGTH = 0,
ITC_BARCODE_FIXED_LENGTH,
ITC BARCODE LENGTH NO CHANGE = 255
} ITC BARCODE LENGTH ID;
```

IS9CConfig::GetTelepen

This function retrieves the current Telepen settings.

Syntax

HRESULT IS9CConfig::GetTelepen(ITC_TELEPEN_DECODING*
 peDecode, ITC_TELEPEN_FORMAT* peFormat);

Parameters

peDecode	[out]	Pointer to the ITC_TELEPEN_DECODING location to receive the decoding for TELEPEN symbology.
peFormat	[out]	Pointer to the ITC_TELEPEN_FORMAT location to receive the format.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

IS9CConfig::SetTelepen

This function updates the Telepen settings with new values.

Syntax

HRESULT IS9CConfig::SetTelepen(ITC_TELEPEN_DECODING* eDecode, ITC TELEPEN FORMAT* eFormat);

Parameters

eDecode [in] Identifies the decoding for Telepen symbology.

eFormat [in] Identifies the format.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

Telepen Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_TELEPEN_DECODING
Format	ASCII	ITC_TELEPEN_FORMAT

Telepen Enumerations

```
typedef enum tagTelepenDecoding
{
ITC_TELEPEN_NOTACTIVE = 0, // Default
ITC_TELEPEN_ACTIVE = 1,
ITC_TELEPEN_NO_CHANGE = 255
} ITC_TELEPEN_DECODING;
typedef enum tagTelepenDecoding
{
ITC_TELEPEN_FORMAT_ASCII, // Default
ITC_TELEPEN_FORMAT_NUMERIC,
ITC_TELEPEN_FORMAT_NO_CHANGE = 255
} ITC_TELEPEN_FORMAT;
```

IS9CConfig::GetUpcEan

This function retrieves the current UPC/EAN settings.

Syntax

HRESULT IS9CConfig::GetUpcEan(ITC_UPCEAN_DECODING* upceanDecode, ITC_UPCA_SELECT* upcASelect, ITC_UPCE_SELECT* upcESelect, ITC_EAN8_SELECT* ean8Select, ITC_EAN13_SELECT* ean13Select, ITC_UPCEAN_ADDON_DIGITS* upcAddOnDigits, ITC_UPCEAN_ADDON_TWO* upcAddOn2, ITC_UPCEAN_ADDON_FIVE* upcAddOn5, ITC_UPCA_CHECK_DIGIT* upcACheck, ITC_UPCE_CHECK_DIGIT* upcECheck, ITC_EAN8_CHECK_DIGIT* ean8Check, ITC_EAN13_CHECK_DIGIT* ean13Check, ITC_UPCA_NUMBER_SYSTEM* upcANumSystem, ITC_UPCE_NUMBER_SYSTEM* upcENumSystem, ITC_UPCA_REENCODE* upcAReencode, ITC_UPCE_REENCODE* upcEReencode, ITC_EAN8_REENCODE* ean8Reencode);

Parameters

upceanDecode	[out]	Pointer to the ITC_UPCEAN_DECODING location to receive the decoding for UPC/EAN symbology.
upcASelect	[out]	Pointer to the ITC_UPCA_SELECT location to receive the UPC-A selection state.
upcESelect	[out]	Pointer to the ITC_UPCE_SELECT location to receive the UPC-E selection state.
ean8Select	[out]	Pointer to the ITC_EAN8_SELECT location to receive the EAN-8 selection state.
ean13Select	[out]	Pointer to the ITC_EAN13_SELECT location to receive the EAN-13 selection state.
upcAddOnDigits	[out]	Pointer to the ITC_UPCEAN_ADDON_DIGITS location to receive the add-on digits.
upcAddOn2	[out]	Pointer to the ITC_UPCEAN_ADDON_TWO location to receive the add-on 2 digits.
upcAddOn5	[out]	Pointer to the ITC_UPCEAN_ADDON_FIVE location to receive the add-on 5 digits.

upcACheck	[out]	Pointer to the ITC_UPCA_CHECK_DIGIT location to receive the UPC-A check digit.
upcECheck	[out]	Pointer to the ITC_UPCE_CHECK_DIGIT location to receive the UPC-E check digit.
ean8Check	[out]	Pointer to the ITC_EAN8_CHECK_DIGIT location to receive the EAN-8 check digit.
ean13Check	[out]	Pointer to the ITC_EAN13_CHECK_DIGIT location to receive the EAN-13 check digit.
upcANumSystem	[out]	Pointer to the ITC_UPCA_NUMBER_SYSTEM location to receive the UPC-A number system.
upcENumSystem	[out]	Pointer to the ITC_UPCE_NUMBER_SYSTEM location to receive the UPC-E number system.
upcAReencode	[out]	Pointer to the ITC_UPCA_REENCODE location to receive the UPC-A reencoding.
upcEReencode	[out]	Pointer to the ITC_UPCE_REENCODE location to receive the UPC-E reencoding.
ean8Reencode	[out]	Pointer to the ITC_EAN8_REENCODE location to receive the EAN-8 reencoding.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

IS9CConfig::SetUpcEan

This function updates the UPC/EAN settings with new values.

Syntax

HRESULT **IS9CConfig::SetUpcEan(** ITC_UPCEAN_DECODING upceanDecode, ITC_UPCA_SELECT_upcASelect, ITC_UPCE_SELECT upcESelect, ITC_EAN8_SELECT_ean8Select, ITC_EAN13_SELECT ean13Select, ITC_UPCEAN_ADDON_DIGITS_upcAddonDigits, ITC_UPCEAN_ADDON_TWO_upcAddon2, ITC_UPCEAN_ADDON_FIVE upcAddOn5, ITC_UPCA_CHECK_DIGIT_upcACheck, ITC_UPCE_CHECK_DIGIT_upcECheck, ITC_EAN8_CHECK_DIGIT ean8Check, ITC_EAN13_CHECK_DIGIT_ean13Check, ITC_UPCA_NUMBER_SYSTEM_upcANumSystem, ITC_UPCE_NUMBER_SYSTEM upcENumSystem, ITC_UPCA_REENCODE_upcAReencode, ITC_UPCE_REENCODE_upcEReencode, ITC_EAN8_REENCODE ean8Reencode);

Parameters

upceanDecode	[in]	Identifies the decoding for UPC/EAN symbology.
upcASelect	[in]	Identifies the UPC-A selection state.
upcESelect	[in]	Identifies the UPC-E selection state.
ean8Select	[in]	Identifies the EAN-8 selection state.
ean13Select	[in]	Identifies the EAN-13 selection state.
upcAddOnDigits	[in]	Identifies the Add-on digits.
upcAddOn2	[in]	Identifies the Add-on 2 digits.
upcAddOn5	[in]	Identifies the Add-on 5 digits.
upcACheck	[in]	Identifies the UPC-A check digit.
upcECheck	[in]	Identifies the UPC-E check digit.
ean8Check	[in]	Identifies the EAN-8 check digit.
ean13Check	[in]	Identifies the EAN-13 check digit.
upcANumSystem	[in]	Identifies the UPC-A number system.
upcENumSystem	[in]	Identifies the UPC-E number system.
upcAReencode	[in]	Identifies the UPC-A reencoding.
upcEReencode	[in]	Identifies the UPC-E reencoding.
ean8Reencode	[in]	Identifies the EAN-8 reencoding.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also None.

Parameter	Default	Valid Range
Decoding	ITC_UPCEAN_NO_CHANGE	This parameter is no longer used, set it to this value.
UPC-A	Active	ITC_UPCA_SELECT
UPC-E	Active	ITC_UPCE_SELECT
EAN-8	Active	ITC_EAN8_SELECT
EAN-13	Active	ITC_EAN13_SELECT
Add On Digits	Not Required	ITC_UPCEAN_ADDON_DIGITS
Add On 2 Digits	Not Active	ITC_UPCEAN_ADDON_TWO
Add On 5 Digits	Not Active	ITC_UPCEAN_ADDON_FIVE
UPC-A Check Digit	Transmitted	ITC_UPCA_CHECK_DIGIT
UPC-E Check Digit	Transmitted	ITC_UPCE_CHECK_DIGIT
EAN-8 Check Digit	Transmitted	ITC_EAN8_CHECK_DIGIT
EAN-13 Check Digit	Transmitted	ITC_EAN13_CHECK_DIGIT
UPC-A Number System	Transmitted	ITC_UPCA_NUMBER_SYSTEM
UPC-E Number System	Transmitted	ITC_UPCE_NUMBER_SYSTEM
Reencode UPC-A	UPC-A transmitted as EAN-13	ITC_UPCA_REENCODE
Reencode UPC-E	UPC-E transmitted as UPC-E	ITC_UPCE_REENCODE
Reencode EAN-8	EAN-8 transmitted as EAN-8	ITC_EAN8_REENCODE

UPC/EAN Default Settings

UPC/EAN Enumerations

	• • • • •	
typedef enum tagUpcEanDecoding		
{		
$ITC_UPCEAN_NOTACTIVE = 0,$		
ITC UPCEAN ACTIVE = $1,$	//	Default
ITC UPCEAN NO CHANGE = 255		
} ITC UPCEAN DECODING;		
typedef enum tagUpcASelect		
{		
ITC UPCA DEACTIVATE,		
ITC UPCA ACTIVATE,	11	Default
ITC UPCA NO CHANGE = 255		
} ITC UPCA SELECT;		
typedef enum tagUpcESelect		
{		
ITC UPCE DEACTIVATE,		
TTC UPCE ACTIVATE.	11	Default
TTC IPCE NO CHANGE = 255	, ,	
TTO TIDOE SELECT.		
tymodof onum tagEan@Soloct		
cypeder enum tagranoserect		
ITC_EAN8_DEACTIVATE,	, ,	
ITC_EAN8_ACTIVATE,	//	Default
$ITC_EAN8_NO_CHANGE = 255$		
} ITC_EAN8_SELECT;		
typedef enum tagEan13Select		
{		
ITC_EAN13_DEACTIVATE,		

ITC EAN13 ACTIVATE, // Default ITC EAN13 NO CHANGE = 255 } ITC EAN13 SELECT; typedef enum tagUpcEanAddonDigits ITC_UPCEAN_ADDON_NOT_REQUIRED, // Default ITC UPCEAN ADDON REQUIRED, ITC_UPCEAN_ADDON_NO_CHANGE = 255 } ITC UPCEAN ADDON DIGITS; typedef enum tagUpcEanAddonTwo ITC UPCEAN ADDON TWO NOTACTIVE = 0, // Default ITC UPCEAN ADDON TWO ACTIVE = 1, ITC UPCEAN ADDON TWO NO CHANGE = 255 } ITC UPCEAN ADDON TWO; typedef enum tagUpcEanAddonFive // Default ITC UPCEAN ADDON FIVE NOTACTIVE = 0, ITC UPCEAN ADDON FIVE ACTIVE = 1, ITC_UPCEAN_ADDON_FIVE_NO_CHANGE = 255 } ITC_UPCEAN_ADDON_FIVE; typedef enum tagUpcACheckDigit ITC UPCA CHECK NOTXMIT = 0, // Default ITC UPCA CHECK XMIT = 1, ITC UPCA CHECK NO CHANGE = 255 } ITC UPCA CHECK DIGIT; typedef enum tagUpcECheckDigit ITC UPCE CHECK NOTXMIT = 0, ITC UPCE CHECK XMIT = 1, // Default ITC UPCE CHECK NO CHANGE = 255 } ITC UPCE CHECK DIGIT; typedef enum tagEan8CheckDigit { ITC EAN8 CHECK NOTXMIT = 0, // Default ITC EAN8 CHECK XMIT = 1, ITC_EAN8_CHECK NO CHANGE = 255} ITC EAN8 CHECK DIGIT; typedef enum tagEan13CheckDigit ITC EAN13 CHECK NOTXMIT = 0, ITC EAN13 CHECK XMIT = 1, // Default ITC EAN13 CHECK NO CHANGE = 255 } ITC EAN13 CHECK DIGIT; typedef enum tagUpcANumberSystem ITC UPCA NUM SYS NOTXMIT = 0, ITC UPCA NUM SYS XMIT = 1, // Default ITC UPCA NUM SYS NO CHANGE = 255 } ITC UPCA NUMBER SYSTEM; typedef enum tagUpcENumberSystem ITC UPCE NUM SYS NOTXMIT = 0, ITC UPCE NUM SYS XMIT = 1, // Default ITC UPCE NUM SYS NO CHANGE = 255 } ITC UPCE NUMBER SYSTEM; typedef enum tagUpcAReencode {

ITC UPCA XMIT AS EAN13, // Default ITC_UPCA_XMIT_AS_UPCA, ITC_UPCA_XMIT_NO_CHANGE = 255 } ITC UPCA REENCODE; typedef enum tagUpcEReencode { // Default ITC_UPCE_XMIT_AS_UPCE, ITC_UPCE_XMIT_AS_UPCA, ITC_UPCE_XMIT_NO_CHANGE = 255 } ITC UPCE REENCODE; typedef enum tagEan8Reencode { ITC EAN8 XMIT AS_EAN8, //Default ITC EAN8 XMIT AS EAN13, ITC_EAN8_XMIT_NO_CHANGE = 255 } ITC_EAN8_REENCODE;

IS9CConfig2 Functions

This interface is derived from the IS9CConfig interface and provides additional methods that can be used to set and retrieve the 700 Series Computer's bar code configuration. All supported symbologies are initialized to their defaults when the S9C firmware is loaded.

GET/SET functions use enumerations as their parameters. In most enumerations, there is an enumerator xx_NO_CHANGE (such as ITC_CODE39_NO_CHANGE), where xx refers to a particular enumeration. This enumerator can be used during a call to a SET to indicate that no change is to be made to that particular parameter. This prevents the called function from having to format the same S9C command and send it down to the scanner.

To specify a bar code length of "any length," use a value of "0" for the bar code length argument.

IS9CConfig2 functions are the following. IS9CCONFIG.H is the header file and ITCUUID.LIB contains the IID_IADC Interface GUID value used to obtain the interface.

- IS9CConfig2::GetCode11 (page 205)
- IS9CConfig2::SetCode11 (page 205)
- IS9CConfig2::GetCustomSymIds (page 207)
- IS9CConfig2::SetCustomSymIds (page 208)
- IS9CConfig2::GetGlobalAmble (page 211)
- IS9CConfig2::SetGlobalAmble (page 212)
- IS9CConfig2::GetPDF417Ext (page 213)
- IS9CConfig2::SetPDF417Ext (page 213)
- IS9CConfig2::GetSymIdXmit (page 214)
- IS9CConfig2::SetSymIdXmit (page 214)

IS9CConfig2::GetCode11

This function retrieves the current settings for Code 11.

Syntax

```
,
HRESULT GetCodel1( ITC_CODE11_DECODING* peDecode,
ITC_CODE11_CHECK_DIGIT* peCheck,
ITC_CODE11_CHECK_VERIFICATION* peVer );
```

Parameters

peDecode	[out]	Pointer to ITC_CODE11_DECODING location to receive Code 11 decoding.
peCheck	[out]	Pointer to ITC_CODE11_CHECK_DIGIT location to receive the check digit option.
peVer	[out]	Pointer to ITC_CODE11_CHECK_VERIFICATION location to receive the check verification option.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

IS9CConfig2::SetCode11

This function updates the current setting of Code 11 symbology.

Syntax

```
HRESULT SetCode11( ITC_CODE11_DECODING eDecode,
ITC_CODE11_CHECK_DIGIT eCheck, ITC_CODE11_CHECK_VERIFICATION
eVer );
```

Parameters

eDecode	[in]	An enumeration that identifies decoding option for Code 11.
eCheck	[in]	An enumeration that identifies the check digit option.
e Ver	[in]	An enumeration that identifies check verification option.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

Code 11 Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_CODE11_DECODING
Check Verification	1 Digit	ITC_CODE11_CHECK_VERIFICATION
Check Digit	Enable	ITC_CODE11_CHECK_DIGIT

Code 11 Enumerations

```
typedef enum tagCode11Decoding
{
ITC CODE11 NOTACTIVE = 0,
ITC CODE11 ACTIVE = 1, // Default
ITC CODE11 NO CHANGE = 255
} ITC CODE11 DECODING;
typedef enum tagCodel1CheckVerification
ITC_CODE11_CHK_VERIFY_ONEDIGIT = 1,
ITC_CODE11_CHK_VERIFY_TWODIGIT = 2, // Default
ITC_CODE11_CHK_VERIFY_NO_CHANGE = 255
} ITC CODE11 CHECK VERIFICATION;
typedef enum tagCode11CheckDigit
{
ITC CODE11 CHECK NOTXMIT = 0, // Default
ITC CODE11 CHECK XMIT = 1,
ITC CODE11 CHECK NO CHANGE = 255
} ITC_CODE11_CHECK_DIGIT;
```

IS9CConfig2::GetCustomSymIds

This function retrieves all the custom symbology identifiers defined for the currently supported symbologies. *This is not supported when using an imager on the 700 Series Computer.*

Syntax

```
HRESULT GetCustomSymIds( ITC_CUST_SYM_ID_PAIR*
pStructSymIdPair,DWORD dwMaxNumElement, DWORD* pdwNumElement
);
```

Parameters

pStructSymIdPair	[out]	Pointer to ITC_CUST_SYM_ID_PAIR location to receive the current defined symbology identifiers for the supported symbologies. The caller must preallocate this buffer with <i>dwMaxNumElement</i> elements.
dwMaxNumElement	[in]	Maximum number of elements allocated for the <i>pStructSymIdPair</i> buffer which should always be equal to the last defined enumeration constant + 1 of the enumeration ITC_CUSTOM_ID. In this case, it is ITC_CUSTOMID_LAST_ELEMENT.
pdwNumElement	[out]	Pointer to DWORD location to receive the actual number of elements returned in the <i>pStructSymIdPair</i> buffer, which should be the same as <i>dwMaxNumElement</i> .

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

- Custom Identifier Assignments (page 209)
- Custom Identifier Example (page 210)
- Custom Identifier Default Settings (page 210)

IS9CConfig2::SetCustomSymIds

This function updates the symbology identifiers (any ASCII values) for the currently supported symbologies. *This is not supported when using an imager on the 700 Series Computer.*

Syntax

```
HRESULT SetCustomSymIds( ITC_CUST_SYM_ID_PAIR*
pStructSymIdPair, DWORD dwNumElement );
```

Parameters

pStructSymIdPair	[in]	Pointer to ITC_CUST_SYM_ID_PAIR location, containing the new symbology identifiers for any supported symbologies to update.
dwNumElement	[in]	Identifies the number of symbology identifiers to update in the <i>pStructSymIdPair</i> buffer.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

Custom Identifier Assignments

Each custom identifier is a one byte ASCII value within the range from 0x00 to 0xff. The enumerations in the ITC_CUSTOM_ID enumerator can be used as symbology identifications in the GetCustomSymIds() and SetCustomSymIds() functions.

Custom	Identifier	Default	Settings
--------	------------	---------	----------

Symbology	Default	Valid Range
Codabar	D	0x00-0xFF
Code 11	*	0x00-0xFF
Code 39	*	0x00-0xFF
Code 93	D	0x00-0xFF
Code128/EAN 128	D	0x00-0xFF
EAN-8	0xFF	0x00-0xFF
EAN-13	F	0x00-0xFF
Interleaved 2 of 5	I	0x00-0xFF
Matrix 2 of 5	D	0x00-0xFF
MSI	D	0x00-0xFF
PDF 417	*	0x00-0xFF
Plessey	D	0x00-0xFF
Standard 2 of 5	D	0x00-0xFF
Telepen	*	0x00-0xFF
UPC-A	А	0x00-0xFF
UPC-E	E	0x00-0xFF

Custom Identifier Example

The following code segment is an example of updating the UPC-E and UPC-A symbology identifiers with new values, and then retrieving the currently defined symbology identifiers for all the supported symbologies:

IS9CConfig2::GetGlobalAmble

This retrieves the scanner's current preamble or postamble setting.

Syntax

HRESULT **GetGlobalAmble(** ITC_GLOBAL_AMBLE_ID *eAmbleId*, BYTE rgbBuffer[], DWORD dwBufferSize, DWORD* pdwBufferSize);

Parameters

eAmbleId	[in]	An enumeration of type ITC_GLOBAL_AMBLE_ID identifies whether the preamble or postamble setting is to be retrieved. Only one setting can be queried at a time.
rgbBuffer	[in]	Contains the buffer for the postamble or preamble setting to be queried.
dwBufferSize	[in]	The maximum number of bytes that rgbBuffer can store. Must be at least ITC_GLOBAL_AMBLE_MAX_CHARS bytes.
pdwBufferSize	[out]	A pointer to DWORD location to store the actual number of returned bytes in <i>rgbBuffer</i> .

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

IS9CConfig2::SetGlobalAmble

This function updates the scanner's current preamble or postamble setting depending on the input parameters.

Syntax

```
HRESULT SetGlobalAmble( ITC_GLOBAL_AMBLE_ID eAmbleId, BYTE
rgbBuffer[], DWORD dwBufferSize );
```

Parameters

eAmbleId	[in]	An enumeration of type ITC_GLOBAL_AMBLE_ID identifies whether the preamble or postamble setting is to be updated. Only one setting can be updated at a time.
rgbBuffer	[in]	Contains the buffer for the postamble or preamble setting to be updated.
dwBufferSize	[in]	Identifies number of bytes in <i>rgbBuffer</i> .

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

Postamble and Preamble Defaults

Parameter	Default	Valid Range
Preamble	Null	0 to 20 ASCII characters
Postamble	Null	0 to 20 ASCII characters

IS9CConfig2::GetPDF417Ext

This function is an extended function for retrieving the PDF 417 settings not included in the IS9CConfig::GetPDF417.

Syntax

HRESULT GetPDF417Ext(ITC_MICRO_PDF417_DECODING* peDecode, ITC_MICRO_PDF417_CODE128_EMULATION* peCode128);

Parameters

peDecode	[out]	Pointer to ITC_MICRO_PDF417_DECODING location to receive the Micro PDF 417 decoding.
peCode128	[out] ITC_	Pointer to _MICRO_PDF417_CODE128_EMULATION* location to receive the Micro PDF 417 Code 128 emulation option.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

IS9CConfig2::SetPDF417Ext

This function is an extended function for updating the additional PDF 417 settings not included in IS9CConfig::SetPDF417.

Syntax

```
HRESULT SetPDF417Ext( ITC_MICRO_PDF417_DECODING eDecode,
ITC MICRO PDF417 CODE128 EMULATION eCode128 );
```

Parameters

eDecode	[in]	An enumeration that identifies decoding option for the Micro PDF 417.
Code128	[in]	An enumeration that identifies the Code 128 emulation

eCode128 [In] An enumeration that identifies the Code 128 emulation option for the Micro PDF 417.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

PDF 417 Extended: Micro PDF 417 Default Settings

Parameter	Default	Valid Range
Decoding	Not Active	ITC_MICRO_PDF417_DECODING
Code 128 Emulation	Not Active	ITC_MICRO_PDF417_CODE128_EMULATION
* These are Micro PDF 417 parameters.		

IS9CConfig2::GetSymIdXmit

This function retrieves the current symbology ID transmission option as described on the next page.

Syntax

HRESULT GetSymIdXmit(ITC SYMBOLOGY ID XMIT* peSymIdXmit);

Parameters

peSymIdXmit [out]

Pointer to ITC_SYMBOLOGY_ID_XMIT location to receive the current symbology identifier transmission option.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

None.

IS9CConfig2::SetSymIdXmit

This updates the symbology ID transmission option shown next page.

Syntax

HRESULT SetSymIdXmit(ITC SYMBOLOGY ID XMIT eSymIdXmit);

Parameters

eSymIdXmit

[in] Identifies the symbology identifier transmission option to update.

Return Values

HRESULT that indicates success or failure.

Remarks

None.

See Also

Symbology ID Transmission Option

The symbology identifier (or code mark) concept provides a standardized way for a device receiving data from a bar code reader to differentiate between the symbologies.

The following symbology ID transmission option specifies whether or not the symbology ID should be transmitted as part of the scanned bar code label to all the connected data collection applications. Options for transmission are: do not transmit, transmit the standard AIM identifiers, or transmit the one byte custom defined identifiers. AIM and custom identifiers cannot be selected to be transmitted at the same time; only the last selected option will be active.

typedef enum tagSymbology	IdXmit
{	
ITC_ID_XMIT_DISABLE = 0	Symbology identifier will not be transmitted as part of the label. This is the default setting.
ITC_ID_XMIT_CUSTOM = 1	Activate custom symbology identifier transmission for all symbologies. Example of the transmitted label: [preamble] [Custom ID] <data> [postamble]</data>
ITC_ID_XMIT_AIM = 2	Activate AIM symbology identifier transmission for all symbologies. Example of the transmitted label: [preamble] [AIM symbology ID] <data> [postamble]</data>

}ITC_SYMBOLOGY_ID_XMIT;

IS9CConfig3 Functions

The IS9CConfig3 interface provides generic methods for retrieving and setting configuration using ISCP commands.

ISCP Commands

An ISCP Command is composed of three or more bytes formatted as <SG><FID><parameters> where:

- *SG* Setup group.
- *FID* Function ID.
- *parameters* One or more configuration value bytes depending on the configuration.

ISCP commands include the following:

Imager Settings

This dictates the start and end column positions for the image dimension.

<u>SG</u>	FID	<u>Parameter</u>	Description
0x7B	80	Value [0639]	Start column position.
0x7B	81	Value [0639]	End column position.

Trigger Settings

This sets the duration of the aiming beam before acquiring images to be decoded.

<u>SG</u>	<u>FID</u>	<u>Parameter</u>	<u>Description</u>
0x70	81	Value [065535]	Number of milliseconds.

QRCode Symbology

This enables or disables the QRCode symbology.

<u>SG</u>	<u>FID</u>	Parameter	<u>Description</u>
0x55	40	0	Disable this symbology.
0x55	40	1	Enable this symbology.

Data Matrix Symbology

This enables or disables the Data Matrix symbology.

<u>SG</u>	FID	<u>Parameter</u>	Description
0x54	40	0	Disable this symbology.
0x54	40	1	Enable this symbology.
ISCP::GetConfig

This retrieves configurations using the ISCP commands format.

Syntax

HRESULT ISCPGetConfig(BYTE rgbCommandBuff[], DWORD dwCommandBuffSize, BYTE rgbReplyBuff[], DWORD dwReplyBuffMaxSize, DWORD *pdwReplyBuffSize);

Parameters

rgbCommandBuff	[in, size_is]	Contains ISCP commands in array of bytes.
dwCommandBuffSize	[in]	Number of bytes in rgbCommandBuff.
rgbReplyBuff	[in, out, size_is]	Results of query in array of bytes.
dwReplyBuffMaxSize	[in]	Maximum size of rgdReplyBuff.
pdwReplyBuffSize	[in, out]	Number of bytes placed in <i>rbfReplyBuff</i> .

Return Values None.

Remarks None.

See Also None.

ISCP::SetConfig

This updates configurations using the ISCP commands format.

Syntax

HRESULT ISCPSetConfig(BYTE rgbCommandBuff[], DWORD dwCommandBuffSize, BYTE rgbReplyBuff[], DWORD dwReplyBuffMaxSize, DWORD *pdwReplyBuffSize);

Parameters

rgbCommandBuff	[in, size_is]	Contains ISCP commands in array of bytes.
dwCommandBuffSize	[in]	Number of bytes in rgbCommandBuff.
rgbReplyBuff	[in, out, size_is]	Results of request in array of bytes.
dwReplyBuffMaxSize	[in]	Maximum size of rgbReplyBuff.
pdwReplyBuffSize	[in, out]	Number of bytes placed in rgbReplyBuff.

Return Values None.

Remarks None.

See Also None.

AIM Symbology ID Defaults

Refer to the official AIM documentation on symbology identifiers for full information on the different processing options supported.

Symbology	ID Character	Modifier Characters
Codabar	F	 Standard Codabar symbol. No special processing. ABC Codabar (American Blood commission) concatenate/message append performed. Reader has validated the check character. Reader has stripped the check character before transmission.
Code 11	Η	 Single modulo 11 check character validated and transmitted. Two modulo 11 check characters validated and transmitted. Check characters validated but not transmitted.
Code 39	Α	 No check character validation nor full ASCII processing. All data transmitted as decoded. Modulo 43 check character validated and transmitted. Modulo 43 check character validated but not transmitted. Full ASCII character conversion performed. No check character validation. Full ASCII character conversion performed. Modulo 43 check character validated and transmitted. Full ASCII character conversion performed. Modulo 43 check character validated and transmitted. Full ASCII character conversion performed. Modulo 43 check character validated but not transmitted.
Code 93	G	0 No options specified. Always transmit 0.
Code128	С	 Standard data packet. No FNC1 in first or second symbol character position after start character. EAN/UCC-128 data packet. FNC1 in first symbol character position after start character. FNC1 in second symbol character position after start character. Concatenation according to International Society for Blood Transfusion specifications was performed. Concatenated data follows.
Interleaved 2 of 5	Ι	 No check character validation. Modulo 10 symbol check character validated and transmitted Modulo 10 symbol check character validated but not transmitted.
Matrix 2 of 5	Х	0 F For symbologies or symbology options not listed, a code character with the value 0-F may be assigned by the decoder manufacturer to identify those symbologies and options implemented in the reader.
MSI	М	 Modulo 10 symbol check character validated and transmitted. Modulo 10 symbol check character validated but not transmitted.

Chapter 6 — Scanner Support

Symbology (continued)	ID Character	Modifier Characters	
PDF 417/ Micro PDF 417	L	 Reader set to conform with protocol defined in 1994 PDF 417 specifications. Reader set to follow protocol of ENV 12925 for Extended Channel Interpretation (all data characters 92 doubled). Reader set to follow protocol of ENV 12925 for Basic Channel Interpretation (data characters 92 are not doubled). Code 128 emulation: implied FNC1 in first position. Code 128 emulation: implied FNC1 after initial letter or pair of digits. Code 128 emulation: no implied FNC1. 	
Plessey	Р	0 No options specified. Always transmit 0.	
Standard 2 of 5 (2-bar start/stop)	R	 No check character validation. Modulo 7 check character validated and transmitted. Modulo 7 check character validated but not transmitted. 	
Standard 2 of 5 (3-bar start/stop)	S	0 No options specified. Always transmit 0.	
Telepen	В	 Full ASCII mode Double density numeric only mode Double density numeric followed by full ASCII Full ASCII followed by double density numeric 	
UPC/EAN	Ε	 Consider UPC/EAN symbols with supplements as two separate symbols. The first symbol is the main data packet, and the second symbol is the 2 or 5 digit supplement. Transmit these two symbols separately, each with its own symbology identifier. Provision is made for the option of transmitting both symbols as a single data packet. 0 Standard data packet in full EAN format (13 digits for EAN-13, UPC-A, and UPC-E; does not include add-on data). 1 Two digit add-on data only. 2 Five digit add-on data only. 3 Combined data packet comprising 13 digits from EAN-13, UPC-A, or UPC-E symbol and 2 or 5 digits from add-on symbol. 4 EAN-8 data packet 	
IMPORTANT : The "symbology_id" character letter must be uppercase for the above definitions.			

IImage Interface

The IImage interface gives the application the capability to acquire images. The image acquired can be either a raw image as captured by the digital camera or it can be normalized. A normalized image is presented the same as if the picture were taken at right angles to the image and at the same distance. The normalized image is commonly used for signature capture applications.

- IImage::ReadSigCapBuffer (page 221)
- IImage::ReadSigCapFile (page 224)
- IImage::ReadImage (page 225)
- IImage::CancelReadImage (page 226)
- IImage::Start (page 226)
- IImage::Stop (page 227)
- IImage::Open (page 227)
- IImage::Close (page 228)

IImage::ReadSigCapBuffer

Syntax

```
HRESULT IImage::ReadSigCapBuffer( ITC_SIGCAP_SPEC
*pSigCapSpec, ITC_IMAGE_SPEC *pImgBuffer, DWORD nMaxBuffSize
);
```

Parameters Parameters: *pSigCapSpec*

[in] Pointer to the structure that identifies the signature capture region. This structure is defined as follows:

 dwStructSize 	Size, in bytes, of this struct. This is for version control.
• iAspectRatio	Ratio of the bar code height (linear bar codes) or row height (2D bar codes) to the narrow element width.
• iOffsetX	Offset in X direction, relative to barcode center. Positive values are right of the bar code, negative values to the left.

• iOffsetY	Offset in Y direction, relative to barcode center. Positive values are higher than the bar code, negative values lower.
• uiWidth	Width of signature capture image region in intelligent bar code units.
• uiHeight	Height of the signature capture image region in in intelligent bar code units.

• *iResolution* Number of pixels per intelligent bar code unit.

// TIFF Binary file

// TIFF Gray Scale

// Raw image

// JPEG image

// Returns data a KIM file

```
• eFormat
```

Ο,

1,

2,

3,

4, 5,

```
ITC_FILE_KIM =
ITC_FILE_TIFF_BIN =
ITC_FILE_TIFF_BIN_GROUP4 =
ITC_FILE_TIFF_GRAY_SCALE =
ITC_FILE_RAW =
ITC_FILE_JPEG =
```

```
• eDepth
```

Number of bits per pixel. Currently, only one (monochrome) or eight (gray-scale) are supported.

Format of the image buffer returned as follows. Currently, only ITC_FILE_RAW is supported.

// TIFF Binary Group 4 compressed

pImgBuffer [out]

```
Pointer to the buffer in which the signature capture image will be put.
```

```
typedef struct tagITCImageSpec {
    DWORD dwStructSize;
    LONG biWidth;
    LONG biHeight;
    WORD biBitCount;
```

```
ITC_FILE_FORMAT eFormat;
DWORD biActualImageSize;
DWORD biMaxImageBytes;
BYTE rgbImageData[1];
```

} ITC_IMAGE_SPEC;

where:

- *dwStructSize* Size, in bytes, of this struct. This is for version control. *biWidth* The width of each row in pixels.
- *biHeight* The number of rows in the image data.
- *biBitCount* The number of bits per pixel.
- *eFormat* Identifies the image format.
 - *biActualImageSize* Total bytes of image data returned.
 - *biMaxImageBytes* Maximum bytes that can be stored in *rgbImageData[]*.
- *rgbImageData* Buffer containing the actual data, for example a 640x480 uses a 307200-byte buffer. The array size of this buffer is arbitrary so do *not* use this structure directly to reserve memory. The actual dimension of the buffer is identified by *biMaxImageBytes*.

Return Values

HRESULT identifying success or error. On error, the following codes will be returned:

- S_OK Image successfully returned.
- ITC_RESULT_ERR_BADREGION_E The specified region is not in the image.
- ITC_RESULT_NO_BC_DECODED_E A bar code has not yet been decoded or the last bar code decoded was not a signature capture symbology.
- ITC_IMGBUFF_TOO_SMALL_E *pImgBuffer* is too small to contain the signature captured image.
- ITC_INV_PARAMETER_E One of the parameters is invalid.
- **S_DEVICE_NOT_OPENED_E** The device had not been opened.

Remarks

ReadSigCapBuffer() will return the image from the last decoded label with dimensions identified by the calling parameter. This signature capture region must include the signature capture bar code. The supported bar codes for signature capture are: PDF 417, Code 128, and Code 39. The caller specifies the width, height, and center of the image to be retrieved. This image is independent of any rotation of the bar code relative to the imager. Thus, if the bar code is decoded with the code itself upside down to the imager, the retrieved image will still be right side up. However, if the specified image is outside the field of view a result code of ITC_RESULT_ERR_BADREGION_E will be returned.

This function uses the dimensions of the last decoded bar code as its coordinate system. Thus, all the parameters describing the image size and position are in units called "Intelligent Bar Code Units." An Intelligent Bar Code Unit is equivalent to the narrow element width of the bar code.

The dimensions of the resulting image can be calculated with this formula:

```
Resulting Width = Specified Width * Specified Resolution
Resulting Height = Specified Height * Specified Resolution
```

See Also

Ilmage::ReadSigCapFile



Note: This has not been implemented as of this publication.

Syntax

```
HRESULT IImage::ReadSigCapFile( ITC_SIGCAP_SPEC
*pSigCapSpec, LPCTSTR pszFileName );
```

Parameters

pSigCapSpec	[in]	Pointer to the structure that identifies the signature
		capture region. See ReadSigCapFile (page 221) for
		a description of this structure.

pszFileName [in] Name of the file in which to copy the image.

Return Values

HRESULT identifying success or error. On error, the following codes will be returned:

- S_OK Image successfully returned.
- ITC_RESULT_ERR_BADREGION_E The specified region is not in the image.
- ITC_RESULT_NO_BC_DECODED_E A bar code has not yet been decoded or the last bar code decoded was not a signature capture symbology.
- ITC_FILE_OPEN_E The file could not be opened.
- ITC_INV_PARAMETER_E One of the parameters is invalid.
- **S_DEVICE_NOT_OPENED_E** The device had not been opened.

Remarks

ReadSigCapFile() will write the image from the last decoded label with dimensions identified by the calling parameter. If the file already exists, its contents will be overwritten.

This signature capture region must include the signature capture bar code. The supported bar codes for signature capture are: PDF 417, Code 128, and Code 39. The caller specifies the width, height, and center of the image to be retrieved. This image is independent of any rotation of the bar code relative to the imager. Thus, if the bar code is decoded with the code itself upside down to the imager, the retrieved image will still be right side up. However, if the specified image is outside the field of view a result code of ITC_RESULT_ERR_BADREGION_E will be returned.

This function uses the dimensions of the last decoded bar code as its coordinate system. Thus, all the parameters describing the image size and position are in units called "Intelligent Bar Code Units". An Intelligent Bar Code Unit is equivalent to the narrow element width of the bar code. The dimensions of the resulting image can be calculated with this formula:

Resulting Width = Specified Width * Specified Resolution Resulting Height = Specified Height * Specified Resolution

See Also

None.

IImage::ReadImage

Syntax

```
HRESULT IImage::Read( ITCFileFormat eFormat, DWORD nDepth,
ITC IMAGE SPEC *pImgBuffer, DWORD dwTimeout );
```

Parameters

eFor	rmat	[in]	Format of the image buffer returned as follows. Currently, only ITC_FILE_RAW is supported.
ITC_FILE_KIM = ITC_FILE_TIFF_BIN = ITC_FILE_TIFF_BIN_GROUP4 ITC_FILE_TIFF_GRAY_SCALE ITC_FILE_RAW = ITC_FILE_JPEG =	0, 1, 2 = 2, 3 = 3, 4, 5,	 	Returns data a KIM file TIFF Binary file TIFF Binary Group 4 compressed TIFF Gray Scale Raw image JPEG image
nDe	epth	[in]	Number of bits per pixel. Currently, only eight (gray-scale) are supported.

pImgBuffer [in/out] Pointer to the buffer containing the image.

dwTimeout [in] Milliseconds to wait for the image to be returned.

Return Values

HRESULT identifying success or error. On error, these will be returned:

• S_OK	Image successfully returned.
• ITC_IMGBUFF_TOO_SMALL_E	<i>pImgBuffer</i> is too small to contain the signature captured image.
• ITC_TIMEOUT_E	Timeout.
• ITC_INV_PARAMETER_E	One of the parameters is invalid.
• S_DEVICE_NOT_OPENED_E	The device had not been opened.

Remarks

The image is returned in *pImgBuffer* in the caller specified format.

See Also

IImage::CancelReadImage

Syntax

HRESULT IImage::CancelReadImage();

Parameters

None.

Return Values

Status code indicating success or failure as follows:

• S_OK

Imager closed.

• S_DEVICE_NOT_OPENED_E The device had not been opened.

Remarks

This function causes a pending image read of IImage::ReadImage() to return immediately with an error status. The purpose of this function is to allow the application to release a thread blocked on the ReadImage() call.

See Also

None.

Ilmage::Start

Syntax

HRESULT IImage::Start();

Parameters

None.

Return Values

Status code indicating success or failure as follows:

• S_OK

Imager started.

• S_DEVICE_NOT_OPENED_E The device had not been opened.

Remarks

This function starts the image continuously capturing images.

See Also

Ilmage::Stop

Syntax

HRESULT **IImage::Stop();**

Parameters

None.

Return Values

Status code indicating success or failure as follows:

• S_OK

Imager started.

- S_IMG_NOT_PRESENT_E Unit does not contain an imager.
 - **S_DEVICE_NOT_OPENED_E** Device had not been opened.

Remarks

This function stops the image continuously capturing images.

See Also

None.

Ilmage::0pen

Syntax

HRESULT IImage:: Open (BOOL fSigCapEnable);

Parameters *fSigCapEnable*

[in] When TRUE, signature capture is enabled. When FALSE, it is disabled. Bar code labels are decoded and images (via IImage::ReadImage) the same.

Return Values

Status code indicating success or failure as follows:

- S_OK Imager opened.
 S_IMG_NOT_PRESENT_E Unit does not contain an imager.
- S_DEVICE_CONTENTION_E Device has already been opened.

Remarks

This function exclusively allocates the imager device so that the other IImage methods can be safely called.

See Also

IImage::Close

Syntax

HRESULT IImage::Close();

Parameters

None.

Return Values

Status code indicating success or failure as follows:

• S_OK

Imager closed.

• **S_DEVICE_NOT_OPENED_E** The device had not been opened.

Remarks

This function releases the imager device so that other applications can open it. An IImage::Release() will also close the imager device.

See Also

Data Collection Configuration



Scanner settings for the 700 Series Computer can be configured via the **Data Collection** control panel applet. From the 700 Series Computer, tap **Start** \rightarrow **Settings** \rightarrow the **System** tab \rightarrow the **Data Collection** icon. See *Appendix A*, "*Control Panel Applets*" for more information about the following parameters. Note that these are in alphabetical order.

- Codabar (page 292)
- Code 11 (page 306)
- Code 128 (*page 295*)
 - Code 128 Options (page 296)
 - Code 128 FNC1 Character (page 297)
- Code 39 (page 290)
- Code 93 (page 294)
 - Code 93 Length (page 294)
- Data Matrix (page 308)
- Interleaved 2 of 5 (page 303)
- Matrix 2 of 5 (page 304)
- MSI (page 299)
- PDF 417 (page 300)
 - Macro PDF (page 300)
 - Micro PDF 417 (page 302)
- Plessey (page 298)
- QR Code (*page 307*)
- Standard 2 of 5 (page 291)
- Telepen (page 305)
- UPC/EAN (page 293)

Tethered Scanner

The Intermec Tethered Scanner feature accepts data from the COM1 port wedges it to the keyboard interface, and allows some ADC. This feature can be enabled or disabled from the Today Screen on the 700 Series Computer.

Enabling and Disabling



On the 700 Series Computer, tap Start \rightarrow Today. Tap the bar code scanner icon in the System Tray *(circled in the following illustration)*. Initially, the bar code scanner icon indicates that this feature is disabled *(shown to the left)*.





• Select **Comm Port Wedge** to send any data, coming into the 700 Series Computer through the COM1 port from an external input device, as keyboard data to an application on the desktop.

For example, if you have Pocket Word running on your 700 Series Computer desktop, information scanned with a scanner connected to the COM1 port will appear in the Word document. If another data collection application is running and is active on the 700 Series Computer, the scanned information will appear in that application.



Note: When Comm Port Wedge is selected, regardless of the data sent by the external input device, you cannot control the device or the data format using any of the Intermec scanner control or data transfer APIs from the SDK or the internal Data Collection software. The external input device is governed by what software it has onboard to tell it how to scan, take pictures, or send the data elsewhere.

- Select 1551/1553 to enable the Sabre 1551E or 1553 Tethered Scanner to scan, then send data as keyboard data. The 1551/1553 Tethered Scanner has software onboard that translates scanned data into characters, so the running/active application does not need to know how to do that. All the scanner control and data transfer APIs will work with the 1551/1553 Tethered Scanner, so you can control the device.
- Select **Disable All** to disable this feature and use the COM1 port for another application, such as ActiveSync. An error message will result if this option were not selected, but this action was attempted. Similarly, if ActiveSync is using the COM1 port, and you select **Comm Port Wedge** or **1551/1553**, an error message will result. See "*Error Message*" on page 232 for more information.

Comm Port Wedge
1551/1553
Disable All
Change Comm Settings

Changing Comm Settings

Tap **Change Comm Settings** to configure the settings for the COM1 port. Current settings are restored after a warm-boot, but are lost after a coldboot. When these settings have not been changed, the **OK** button is disabled (grayed out). When changes are made, tap **OK** after it is enabled to accept these changes.

- Baud Rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
- Data Bits: 7 or 8
- Parity: None, Odd, Even, Mark, Space
- **Stop Bits**: 1 or 2
- Flow Control: None or Hardware

Tethered Scanner

The default settings for the Tethered Scanner are shown in the following illustration:

Comm Port 1 Settings		
Baud Rate:	38400 🔻	
Data Bits	8 🔻	
Parity:	None 🔻	
Stop Bits:	1 •	
Flow Control:	None 🔻	

Sabre 1551E or 1553 Tethered Scanner

The default communication configuration for the Sabre 1551E or 1553 Tethered Scanner is shown in the following illustration. Scan the EasySet Reset Factory Defaults label to set the Sabre 1551E or 1553 tethered scanner communications settings to this configuration. The COM1 port configuration settings must also match those of the scanner to scan labels.

Comm Port 1 Settings			
Baud Rate:	9600 🔻		
Data Bits	7 🔻		
Parity:	Even 🔻		
Stop Bits:	2 🔻		
Flow Control:	None 🔻		

Welch Allyn 1470 Imager Settings

The Welch Allyn 1470 Imager can be set to this configuration by scanning the Factory Default Settings label.

Error Message

If the COM1 port is used by another application, such as ActiveSync, neither the Comm Port Wedge nor the 1551/1553 Tethered Scanner can be enabled. As a result, the following message may appear. *Note that this message is for the Comm Port Wedge.* You must disable that application to free up the COM1 port before you can enable either the wedge or the scanner.



Error enabled CommPortWedge: Comm port unavailable.

Scanner Cabling

A null modem cable is required for the Welch Allyn 1470 Imager to communicate with the 700 Series Computer when using the 700 Series Serial Cable (P/N: 226-999-001).

The Sabre 1551E / 1553 Cable connects directly to the Model 700 Comm Port.

Limitations and Capabilities

The Tethered Scanner has the following limitations:

- No auto detection of a scanner's physical connection to COM1 port. User needs to ensure the communication settings of COM1 port matched the settings of the device.
- The Pocket PC Pocket Office applications misbehave when control characters such as carriage return are wedged. This is a known Pocket PC problem, which is being worked with Microsoft and for which a work around is being developed.
- Communications port is COM1 and cannot be changed.
- A complete bar code label is detected when the time between bytes (the inter-byte gap) exceeds 100 ms. This allows that data could be concatenated if two labels were received while the Comm Port Wedge or the 1551/1553 Tethered Scanner was not performing a read. That is, it could be wedging data just read or the read thread could be preempted. Also, the labels could appear concatenated if the scanner itself were to buffer the labels before transmitting them.

When enabled, the Comm Port Wedge menu option has the following limitation:

• There is no bar code API to get bar code data from the bar code scanner. The Comm Port Wedge transmits the data through the keyboard interface only.

When enabled, the 1551/1553 menu option has the following capabilities:

- Grid Data Editing is available.
- The source of the symbology configurations is only available via the Easy Set command labels. Only the Virtual Wedge configurations can be configured via the Data Collection Control Panel Applet Virtual Wedge page. See Appendix A, "*Control Panel Applets*," for more information.
- May transmit the data through the keyboard interface (via the Virtual Wedge).

• The bar code APIs, defined in the IADC interface, are available to get bar code data from the bar code scanner. The following example shows how to programmatically collects bar code data:

```
#include "IADC.h"
                                        // Linked with ITCUUID.LIB
#include ``ITCAdcMgmt.h"
                                       // Linked with ITCAdcDevMgmt.lib
  IADC* pIADC;
  HRESULT hrStatus = S OK;
// Create a ADC COM interface to collect bar code data from the 1551E/1553
// when the 1551/1553 menu option is enabled.
  hrStatus =
  ITCDeviceOpen(TEXT("ExtScanner"), // Name of the ADC device.
    IID_IADC, // COM interface to return
ITC_DHDEVFLAG_READAHEAD, // Device's Flags
(LPVOID *) &pIADC); // the returned interface
if( SUCCEEDED(hrStatus) )
  {
    BYTE byteBuffer[MAX LABEL SIZE];
    DWORD dwLength = 0;
  HRESULT hr = pIDC -> Read(
    byteBuffer,
                                       // Buffer to put the ADC data.
    MAX LABEL SIZE,
                                       // Size of pDataBuffer in bytes.
    &dwLength,
                                       // Number bytes returned.
                                       // Time stamp of the received data. NULL.
    NULL,
    INFINITE
                                       // Number of milliseconds to wait.
  );
}
    when done using this COM interface, delete it:
ITCDeviceClose( (IUnknown **) pIADC);
```



The following programming information pertains to the 700 Series Color Mobile Computer:

- Creating CAB Files (page 236)
- FTP Server (page 251)
- Full Screen (page 262)
- Kernel I/O control functions (page 264)
- Reboot Functions (page 280)
- Remapping the Keypad (page 281)

Creating CAB Files

The Windows CE operating system uses a .CAB file to install an application on a Windows CE-based device. A .CAB file is composed of multiple files that are compressed into one file. Compressing multiple files into one file provides the following benefits:

- All application files are present.
- A partial installation is prevented.
- The application can be installed from several sources, such as a desktop computer or a Web site.

Use the CAB Wizard application (CABWIZ.EXE) to generate a .CAB file for your application.

Creating Device-Specific CAB Files

Do the following to create a device-specific .CAB file for an application, *in the order provided*:

- 1 Create an .INF file with Windows CE-specific modifications (page 236).
- **2** *Optional* Create a SETUP.DLL file to provide custom control of the installation process (*page 248*).
- **3** Use the CAB Wizard to create the .CAB file, using the .INF file, the optional SETUP.DLL file, and the device-specific application files as parameters (*page 249*).

Creating an .INF File

An .INF file specifies information about an application for the CAB Wizard. Below are the sections of an .INF file:

[Version]

This specifies the creator of the file, version, and other relevant information.

Required? Yes

- Signature: "signature_name" Must be "\$Windows NT\$" as Windows CE is not available on Windows 95.
- **Provider**: *"INF_creator"* The company name of the application, such as "Microsoft."
- CESignature: "\$Windows CE\$"

EXAMPLE:

```
[Version]
Signature = "$Windows NT$"
Provider = "Microsoft"
CESignature = "$Windows CE$"
```

[CEStrings]

This specifies string substitutions for the application name and the default installation directory.

Required? Yes

- AppName: *app_name* Name of the application. Other instances of %AppName% in the .INF file will be replaced with this string value, such as RP32.
- InstallDir: *default_install_dir* Default installation directory on the device. Other instances of %Install-Dir% in the .INF file will be replaced with this string value. Example: \storage_card\%AppName%

EXAMPLE:

[CEStrings] AppName="Game Pack" InstallDir=%CE1%\%AppName%

[Strings]

This section is optional and defines one or more string keys. A string key represents a string of printable characters.

Required? No

• string_key: value

String consisting of letters, digits, or other printable characters. Enclose *value* in double quotation marks """" if the corresponding string key is used in an item that requires double quotation marks. No string_keys is okay.

EXAMPLE:

[Strings]

reg_path = Software\Microsoft\My Test App

[CEDevice]

Describes the platform for the targeted application. All keys in this section are optional. If a key is nonexistent or has no data, Windows CE does not perform any checking with the exception being *UnsupportedPlatforms*. If the *UnsupportedPlatforms* key exists but no data, the previous value is not overridden.

Required? Yes

• **ProcessorType** : *processor_type*

The value that is returned by SYSTÉMINFO.dwProcessorType. For example, the value for the SH3 CPU is 10003 and the MIPS CPU is 4000.

• UnsupportedPlatforms: platform_family_name

This lists known unsupported platform family names. If the name specified in the [CEDevice.xxx] section is different from that in the [CEDevice] section, both *platform_family_name* values are unsupported for the microprocessor specified by xxx. That is, the list of unsupported platform family names is appended to the previous list of unsupported names. Application Manager will not display the application for an unsupported platform. Also, a user will be warned during the setup process if the .CAB file is copied to an unsupported device.

EXAMPLE:

[CEDevice]

UnsupportedPlatforms = pltfrm1 ; pltfrm1 is unsupported
[CEDevice.SH3]
UnsupportedPlatforms = ; pltfrm1 is still unsupported

• VersionMin: *minor_version* Numeric value returned by OSVERSIONINFO.dwVersionMinor. The .CAB file is valid for the currently connected device if the version of this device is greater than or equal to VersionMin. For Windows CE Japanese language devices, set this to 2.01

VersionMax: *major_version* Numeric value returned by OSVERSIONINFO.dwVersionMajor. The .CAB file is valid for the currently connected device if the version of this device is less than or equal to VersionMax. For Windows CE Japanese language devices, set this to 2.01



Note: Supported Windows CE operating system versions include 1.0, 1.01, 2.0, 2.01, and 2.10. When using these numbers, be sure to include all significant digits.

• BuildMin: *build_number* Numeric value returned by OSVERSIONINFO.dwBuildNumber. The .CAB file is valid for the currently connected device if the version of this device is greater than or equal to BuildMin.

• BuildMax: *build_number* Numeric value returned by OSVERSIONINFO.dwBuildNumber. The .CAB file is valid for the currently connected device if the version of this device is less than or equal to BuildMax.

EXAMPLE:

The following code example shows three [CEDevice] sections: one that gives basic information for any CPU and two that are specific to the SH3 and the MIPS microprocessors.

```
[CEDevice]
                                ; A "template" for all platforms
UnsupportedPlatforms = pltfrm1 ; Does not support pltfrm1
; The following specifies version 1.0 devices only.
VersionMin = 1.0
VersionMax = 1.0
[CEDevice.SH3]
                               ; Inherits all [CEDevice] settings
; This will create a .CAB file specific to SH3 devices.
ProcessorType = 10003 ; SH3 .cab file is valid for SH3 microprocessors.
UnsupportedPlatforms = ; pltfrm1 is still unsupported
; The following overrides the version settings so that no version checking is
performed.
VersionMin =
VersionMax =
[CEDevice.MIPS]
                               ; Inherits all [CEDevice] settings
; This will create a .CAB file specific to "MIPS" devices.
ProcessorType = 4000 ; MIPS .CAB file is valid for MIPS microprocessor.
UnsupportedPlatforms =pltfrm2 ; pltfrm1, pltfrm2 unsupported for MIPs .CAB file.
```

```
P
```

Note: To create the two CPU-specific .CAB files for the SETUP.INF file in the previous example, run the CAB Wizard with the "/cpu sh3 mips" parameter.

[DefaultInstall]

This describes the default installation of your application. Note that under this section, you will list items expanded upon later in this description.

Required? Yes

• Copyfiles: *copyfile_list_section* Maps to files defined later in the .INF file, such as Files.App, Files.Font, and Files.Bitmaps.

- AddReg: *add_registry_section* Example: RegSettings.All
- CEShortcuts: *shortcut_list_section* String that identifies one more section that defines shortcuts to a file, as defined in the [CEShortcuts] section.
- CESetupDLL: *setup_DLL* Optimal string that specifies a SETUP.DLL file. It is written by the Independent Software Vendor (ISV) and contains customized functions for operations during installation and removal of the application. The file must be specified in the [SourceDisksFiles] section.
- CESelfRegister: *self_reg_DLL_filename* String that identifies files that self-register by exporting the DllRegister-Server and DllUnregisterServer Component Object Model (COM) functions. Specify these files in the [SourceDiskFiles] section. During installation, if installation on the device fails to call the file's exported DllRegisterServer function, the file's exported DllUnregisterServer function will not be called during removal.

EXAMPLE:

[DefaultInstall] AddReg = RegSettings.All CEShortcuts = Shortcuts.All

[SourceDiskNames]

This section describes the name and path of the disk on which your application resides.

Required? Yes

- disk_ordinal: disk_label,, path 1=, "App files", C:\Appsoft\RP32\... 2=, "Font files",, C:\RpTools\... 3=, "CE Tools",, C:\windows ce tools...
- CESignature: "\$Windows CE\$"

Example

[SourceDisksNames]	;	Required section
<pre>1 = ,"Common files",,C:\app\common</pre>	;	Using an absolute path
[SourceDisksNames.SH3]		
2 = ,"SH3 files",,sh3	;	Using a relative path
[SourceDisksNames.MIPS]		
2 = , "MIPS files",,mips	;	Using a relative path

[SourceDiskFiles]

This describes the name and path of the files in which your application resides.

Required? Yes

• filename: disk_number[,subdir] RPM.EXE = 1,c:\appsoft\... WCESTART.INI = 1 RPMCE212.INI = 1 TAHOMA.TTF = 2

Note: [,subdir] is relative to the location of the INF file.

Example

```
[SourceDisksFiles] ; Required section
begin.wav = 1
end.wav = 1
sample.hlp = 1
[SourceDisksFiles.SH3]
sample.exe = 2 ; Uses the SourceDisksNames.SH3 identification of 2.
[SourceDisksFiles.MIPS]
sample.exe = 2 ; Uses the SourceDisksNames.MIPS identification of 2.
```

[DestinationDirs]

This describes the names and paths of the destination directories for the application on the target device. *Note Windows CE does not support directory identifiers.*

Required? Yes

• file_list_section: 0, subdir

String that identifies the destination directory. The following list shows the string substitutions supported by Windows CE. These can be used only for the beginning of the path. \

- %CE1% \Program Files
- %CE2% \Windows
- %CE3% \My Documents
- %CE4% \Windows\Startup
- %CE5% \My Documents
- %CE6% \Program Files\Accessories
- %CE7% \Program Files\Communication
- %CE8% \Program Files\Games
- %CE9% \Program Files\Pocket Outlook
- %CE10% \Program Files\Office
- %CE11% \Windows\Start Menu\Programs
- %CE12% \Windows\Start Menu\Programs\Accessories
- %CE13% \Windows\Start Menu\Programs\Communications
- %CE14% \Windows\Start Menu\Programs\Games
- %CE15% \Windows\Fonts
- %CE16% \Windows\Recent
- %CE17% \Windows\Start Menu
- %InstallDir%

Contains the path to the target directory selected during installation. It is declared in the [CEStrings] section N = N

%AppName%

Contains the application name defined in the [CEStrings] section.

Example

Files.Common	= 0,%CE1%\My Subdir	;	\Program	Files\My	Subdir
Files.Shared	= 0,%CE2%	;	\Windows		

[DestinationDirs]

[CopyFiles]

This section, under the [DefaultInstall] section, describes the default files to copy to the target device. Within the [DefaultInstall] section, files were listed that must be defined elsewhere in the INF file. This section identifies that mapping and may contain flags.

Required? Yes

- copyfile_list_section: destination_filename,[source_filename] The source_filename parameter is optional if it is the same as destination_filename.
- copyfile_list_section: *flags* The numeric value that specifies an action to be done while copying files. The following table shows values supported by Windows CE.

Flag	Value	Description
COPYFLG_WARN_IF_SKIP	0x00000001	Warn user if skipping a file is attempted after error.
COPYFLG_NOSKIP	0x00000002	Do not allow a user to skip copying a file.
COPYFLG_NO_OVERWRITE	0x00000010	Do not overwrite files in destination directory.
COPYFLG_REPLACEONLY	0x00000400	Copy the source file to the destination directory only if the file is already in the destination directory.
CE_COPYFLG_NO_DATE_DIALOG	0x20000000	Do not copy files if the target file is newer.
CE_COPYFLG_NODATECHECK	0x40000000	Ignore date while overwriting the target file.
CE_COPYFLG_SHARED	0x80000000	Create a reference when a shared DLL is counted.

Example
[DefaultInstall.SH3]
CopyFiles = Files.Common, Files.SH3
[DefaultInstall.MIPS]
CopyFiles = Files.Common, Files.MIPS

[AddReg]

This section, under the [DefaultInstall] section, is optional and describes the keys and values that the .CAB file adds to the device registry. Within the [DefaultInstall] section, a reference may have been made to this section, such as "AddReg=RegSettings.All". This section defines the options for that setting.

Required? No

- add_registry_section: registry_root_string String that specifies the registry root location. The following list shows the values supported by Windows CE.
 - HKCR Same as HKEY_CLASSES_ROOT
 - HKCU Same as HKEY_CURRENT_USER
 - HKLM Same as HKEY LOCAL MACHINE
- add_registry_section: value_name Registry value name. If empty, the "default" registry value name is used.
- add_registry_section: *flags* Numeric value that specifies information about the registry key. The following table shows the values that are supported by Window CE.

Flag	Value	Description
FLG_ADDREG_NOCLOBBER	0x00000002	If the registry key exists, do not overwrite it. Can be used with any of the other flags in this table.
FLG_ADDREG_TYPE_SZ	0x00000000	REG_SZ registry data type.
FLG_ADDREG_TYPE_MULTI_SZ	0x00010000	REG_MULTI_SZ registry data type. Value field that follows can be a list of strings separated by commas.
FLG_ADDREG_TYPE_BINARY	0x00000001	REG_BINARY registry data type. Value field that follows must be a list of numeric values separated by commas, one byte per field, and must not use the 0x hexadecimal prefix.
FLG_ADDREG_TYPE_DWORD	0x00010001	REG_DWORD data type. The noncompatible format in the Win32 Setup .INF documentation is supported.

Example

AddReg = RegSettings.All

[RegSettings.All]

HKLM,%reg_path%,,0x0000000,alpha ; <default> = "alpha" HKLM,%reg_path%,test,0x00010001,3 ; Test = 3 HKLM,%reg_path%\new,another,0x00010001,6 ; New\another = 6

[CEShortCuts]

This section, a Windows CE-specific section under the [**DefaultInstall**] section, is optional and describes the shortcuts that the installation application creates on the device. Within the [**DefaultInstall**] section, a reference may have been made to this section, such as "ShortCuts.All". This section defines the options for that setting.

Required? No

- shortcut_list_section: shortcut_filename String that identifies the shortcut name. It does not require the .LNK extension.
- shortcut_list_section: shortcut_type_flag Numeric value. Zero or empty represents a shortcut to a file; any nonzero numeric value represents a shortcut to a folder.
- shortcut_list_section: target_file_path
 String value that specifies the destination location. Use the target file
 name for a file, such as MyApp.exe, that must be defined in a file copy
 list. For a path, use a file_list_section name defined in the [Destination-Dirs] section, such as DefaultDestDir, or the %InstallDir% string.
- shortcut_list_section: standard_destination_path Optional string value. A standard %CEx% path or %InstallDir%. If no value is specified, the shortcut_list_section name of the current section or the DefaultDestDir value from the [DestinationDirs] section is used.

Example

CEShortcuts = Shortcuts.All
[Shortcuts.All]
Sample App,0,sample.exe ; Uses the path in DestinationDirs. Sample
App,0,sample.exe,%InstallDir% ; The path is explicitly specified.

Sample .INF File

[Version] ; Required section Signature = "\$Windows NT\$" Provider = "Intermec Technologies Corporation" CESignature = "\$Windows CE\$"

;[CEDevice]

;ProcessorType =

[DefaultInstall] ; Required section

CopyFiles = Files.App, Files.Fonts, Files.BitMaps, Files.Intl, Files.TelecomNcsCE, Files.Windows, Files.Import, Files.Export, Files.Work, Files.Database, Files.WinCE AddReg = RegSettings.All ;CEShortcuts = Shortcuts.All

[SourceDisksNames] ; Required section

1 = , "App files" ,,c:\appsoft\... 2 = , "Font files" ,,c:\WinNT\Fonts 3 = , "CE Tools" ,,c:\Windows ce tools\wce212\6110ie\mfc\lib\x86

[SourceDisksFiles] ; Required section
rpm.exe = 1,C:\Appsoft\program\wce212\WCEX86Rel6110
wcestart.ini = 1

```
rpmce212.ini = 1
intermec.bmp = 1
rpmlogo.bmp = 1
rpmname.bmp = 1
import.bmp = 1
export.bmp = 1
clock.bmp = 1
printer.bmp = 1
filecopy.bmp = 1
readme.txt = 1
lang eng.bin = 1
rpmdata.dbd = 1, database\wce1
tahoma.ttf = 2
mfcce212.dll = 3
olece212.dll = 3
olece211.dll = 1,c:\windows ce tools\wce211\NMSD61102.11\mfc\lib\x86
rdm45wce.dll = 1,c:\rptools\rdm45wce\4 50\lib\wce212\wcex86rel
picfmt.dll = 1,c:\rptools\picfmt\1 00\wce212\wcex86rel6110
fmtctrl.dll = 1, c: rptools fmtctrl 1 00 wce212 wcex86 rel6110
ugrid.dll = 1,c:\rptools\ugrid\1_00\wce212\wcex86rel6110
simple.dll = 1,c:\rptools\pspbm0c\1 00\wce211\wcex86rel
psink.dll = 1,c:\rptools\psink\1 00\wce211\WCEX86RelMinDependency
pslpwce.dll =1,c:\rptools\pslpm0c\1 00\wce211\WCEX86RelMinDependency
npcpport.dll = 1,c:\rptools\cedk\212 03\installable drivers\printer\npcp
;dexcom.dll = 1,c:\rptools\psdxm0c\1 00\x86
ncsce.exe = 1,c:\rptools\ncsce\1 04
nrinet.dll = 1, c:\rptools\ncsce\sqrt{1} 04
[DestinationDirs] ; Required section
;Shortcuts.All = 0,%CE3% ; \Windows\Desktop
Files.App = 0,%InstallDir%
Files.DataBase= 0,%InstallDir%\DataBaseFiles.BitMaps= 0,%InstallDir%\BitmapsFiles.Fonts= 0,%InstallDir%\FontsFiles.Intl= 0,%InstallDir%\Intl
Files.TelecomNcsCE = 0,%InstallDir%\Telecom\NcsCE
                                0,%installDir%\Windows
0,%InstallDir%\Import
0,%InstallDir%\Export
0,%InstallDir%\Export
0,%InstallDir%\Windows
0,%InstallWindows
Files.Windows = 0,%InstallDir%\Windows
Files.Import
Files.Export
Files.Work
Files.WinCE
                                      = 0,\storage card\wince
[CEStrings]
                                       ; Required section
AppName = Rp32
InstallDir = \storage card\%AppName%
                                  ; Optional section
[Strings]
;[Shortcuts.All]
;Sample App, 0, sample.exe
                                                                                        ; Uses the path in DestinationDirs.
;Sample App,0,sample.exe,%InstallDir% ; The path is explicitly specified.
[Files.App]
rpm.exe,,,0
rpm.ini, rpmce212.ini,,0
mfcce212.dll,,,0
olece212.dll,,,0
olece211.dll,,,0
rdm45wce.dll,,,0
picfmt.dll,,,0
```

```
fmtctrl.dll,,,0
ugrid.dll,,,0
simple.dll,,,0
psink.dll,,,0
pslpwce.dll,,,0
npcpport.dll,,,0
;dexcom.dll,,,0
[Files.DataBase]
rpmdata.dbd,,,0
[Files.Fonts]
tahoma.ttf,,,0
[Files.BitMaps]
intermec.bmp,,,0
rpmlogo.bmp,,,0
rpmname.bmp,,,0
import.bmp,,,0
export.bmp,,,0
clock.bmp,,,0
printer.bmp,,,0
filecopy.bmp,,,0
[Files.Intl]
lang_eng.bin,,,0
[Files.TelecomNcsCE]
ncsce.exe,,,0
nrinet.dll,,,0
[Files.Windows]
readme.txt,,,0
[Files.Import]
readme.txt,,,0
[Files.Export]
readme.txt,,,0
[Files.Work]
readme.txt,,,0
[Files.WinCE]
wcestart.ini,,,0
[ReqSettings.All]
HKLM, "SOFTWARE\Microsoft\Shell\AutoHide",,0x00010001,1
```

```
; Autohide the taskbar HKLM, "SOFTWARE\Microsoft\Shell\OnTop",,0x00010001,0
; Shell is not on top HKLM, "SOFTWARE\Microsoft\Clock",SHOW_CLOCK,0x00010001,0
; Clock is not on taskbar
```

Using Installation Functions in SETUP.DLL

SETUP.DLL is an optional file that enables you to perform custom operations during installation and removal of your application. The following list shows the functions that are exported by SETUP.DLL.

• Install_Init

Called before installation begins. Use this function to check the application version when reinstalling an application and to determine if a dependent application is present.

• Install_Exit

Called after installation is complete. Use this function to handle errors that occur during application installation.

• Uninstall_Init

Called before the removal process begins. Use this function to close the application, if the application is running.

• Uninstall_Exit

Called after the removal process is complete. Use this function to save database information to a file and delete the database and to tell the user where the user data files are stored and how to reinstall the application.



Note; Use [DefaultInstall] \rightarrow CESelfRegister (page 240) in the .INF file to point to SETUP.DLL.

After the CAB File Extraction

Cab files that need to cause a warm reset after cab extraction will need to create the __RESETMEPLEASE__.TXT file in the "\Windows" directory. The preferred method to create this file is within the DllMain portion of the SETUP.DLL file. It looks like this:

```
BOOL APIENTRY DllMain ( HANDLE hModule, DWORD ul reason for call, LPVOID
lpReserved )
  switch (ul reason for call)
     {
        case DLL PROCESS ATTACH:
          break;
        case DLL THREAD ATTACH:
          break;
        case DLL THREAD DETACH:
          break;
        case DLL PROCESS DETACH:
          if (bInstallSuccessful) {
             HANDLE h;
             h = CreateFile(L"\\Windows\\__resetmeplease__.txt",
GENERIC_READ|GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
                FILE ATTRIBUTE HIDDEN, NULL);
             if (h = INVALID HANDLE VALUE)
                CloseHandle(h);
          }
          break;
  }
  return TRUE;
}
```

The system software looks for the following directory structure and files on the installed media card whether it be an SD card or CF card or embedded flash file system. No other folders need exist.

\2577\autorun.exe
\2577\autorun.dat
\2577\autocab.exe
\2577\autocab.dat
\cabfiles*.cab

Creating CAB Files with CAB Wizard

After you create the .INF file and the optional SETUP.DLL file, use the CAB Wizard to create the .CAB file. The command-line syntax for the CAB Wizard is as follows:

```
cabwiz.exe ``inf_file" [/dest dest_directory] [/err error_file] [/cpu cpu_type
[cpu_type]]
```

A batch file, located in <program> directory, with the following commands, works well:

```
cd\"Windows CE Tools"\WCE211\"MS HPC Pro"\support\appinst\bin
cabwiz.exe c:\appsoft\<program>\<inf_file_name>
cd \appsoft\<program>
```

- "inf_file" The SETUP.INF file path.
- *dest_directory*

The destination directory for the .CAB files. If no directory is specified, the .CAB files are created in the "inf_file" directory.

• error_file

The file name for a log file that contains all warnings and errors that are encountered when the .CAB files are compiled. If no file name is specified, errors are displayed in message boxes. If a file name is used, the CAB Wizard runs without the user interface (UI); this is useful for automated builds.

• cpu_type

Creates a .CAB file for each specified microprocessor tag. A microprocessor tag is a label used in the Win32 SETUP.INF file to differentiate between different microprocessor types. The */cpu* parameter, followed by multiple *cpu_type* values, must be the last qualifier in the command line.

Example

This example creates .CAB files for the SH3 and MIPS microprocessors, assuming that the Win32 SETUP.INF file contains the SH3 and MIPS tags:

cabwiz.exe "c:\myfile.inf" /err myfile.err /cpu sh3 mips



Note: CABWIZ.EXE, MAKECAB.EXE, and CABWIZ.DDF (Windows CE files available on the Windows CE Toolkit) must be installed in the same directory on the desktop computer. Call CABWIZ.EXE using its full path for the CAB Wizard application to run correctly.

Troubleshooting the CAB Wizard

To identify and avoid problems that might occur when using the CAB Wizard, follow these guidelines:

- Use %% for a percent sign (%) character when using this character in an .INF file string, as specified in Win32 documentation. This will not work under the [Strings] section.
- Do not use .INF or .CAB files created for Windows CE to install applications on Windows-based desktop platforms.
- Ensure the MAKECAB.EXE and CABWIZ.DDF files, included with Windows CE, are in the same directory as CABWIZ.EXE.
- Use the full path to call CABWIZ.EXE.
- Do not create a .CAB file with the MAKECAB.EXE file included with Windows CE. You must use CABWIZ.EXE, which uses MAKE-CAB.EXE to generate the .CAB files for Windows CE.
- Do *not* set the read-only attribute for .CAB files.

FTP Server

FTP support is provided through the FTP Server application FTPDCE.EXE (MS Windows CE Versions) which is provided as part the base system.

FTPDCE is the Internet File Transfer Protocol (FTP) server process. The server can be invoked from an application or command line. Besides servicing FTP client requests the FTP Server also send a "network announcement" to notify prospective clients of server availability.

Synopsis

ftpdce [options]

Options

• -Aaddr

Sets the single target address to which to send the network announcement. *Default is broadcast.*

• -Bbyte

Sets the FTP data block size. Smaller sizes may be useful over slower links. *Default is 65536*.

• -Cname

Sets the device name. Used by Intermec management software.

• -Fvalue

Disables the default Intermec account. A value of "0" disables the account. *Default is "1"*.



Note: Disabling the default account without providing a working access control list on the server will result in a device that will not accept any FTP connections.

• -Hsec

Sets the interval between network announcements in seconds. A value of "0" turns the network announcement off. *Default is 30 seconds.*

• -*Iip*

Sets the preferred 6920 Communications Server (optional).

• -Llog

Sets the state of logging. Default is 0 (disabled).

• -Nsec

Specifies the number of seconds to wait before starting FTP server services.

- *-Pport* Sets the UDP port on which the network announcement will be sent. *Default port is 52401.*
- -*Qport* Sets the port on which the FTP Server will listen for connections. *Default port is 21.*

• -Rdir

Sets the FTP mount point to this directory. Default is the rootdirectory of the drive from which the FTP Server program was executed.

```
• -Tscript
```

Sets the script name for the 6920 Communications Server to process.

- *-Uurl* Sets the default URL for this device.
- -Z"parms"

Sets extended parameters to be included in the network announcement.

Configurable Parameters Via the Registry Editor

The following parameters receive default values during the installation of the Intermec FTP Server components. A few of the parameters are visible in the registry by default, but most must be created in order to modify the default behavior of the FTP server.

BlockSize

Setting this parameter forces the Intermec FTP Server to transmit and receive Ethernet packets using the specified data block size. By default, the FTP server transmits and receives data using a 64K data block size. Adjusting this value may be useful in certain wireless TCP/IP installations.

Key

HKLM\Software\Intermec\IFTP

Value Type

REG_DWORD - data block size, in bytes.

Valid Range

0x100-0x10000 (256-65536 decimal).

Default

65536
DeviceName

This parameter forces the Intermec FTP Server to include the specified device name in the Intermec Device Network Announcement (IDNA). Adjusting this value may be useful in assigning a symbolic name to this device for asset tracking.

Key

HKLM\Software\Intermec\IFTP

Value Type

REG_SZ

Valid Range

None.

Default

None.

DeviceURL

This parameter forces the Intermec FTP Server to transmit the specified URL in the IDNA. This can be used by Intermec management software for asset management.

Key

HKLM\Software\Intermec\IFTP

Value Type REG_SZ

. . . . -

Valid Range None.

1 10110.

Default

None.

IDNATarget

This parameter forces the Intermec FTP Server to transmit the IDNA to a specific destination instead of a general UDP broadcast. This parameter is useful on networks that do not allow UDP broadcasts to be routed between subnets. The use of this parameter will restrict the reception of the IDNA to the target destination only.

Key

HKLM\Software\Intermec\IFTP

Value Type

REG_SZ

Valid Range

None.

Default

None.

ManifestName

This parameter forces the Intermec FTP Server to transmit the specified manifest name in the IDNA. This parameter is used by the Intermec 6920 Communications Server for communication transactions. See the 6920 Communications Server documentation for proper use of this parameter.

Key

HKLM\Software\Intermec\IFTP

Value Type

REG_SZ

Valid Range

None.

Default

iftp.ini

PauseAtStartup

This parameter forces the Intermec FTP Server to sleep for the specified number of seconds before making the FTP service available on the device.

Key

HKLM\Software\Intermec\IFTP

Value Type REG_DWORD - stored in seconds.

Valid Range

None.

Default

0

Root

This parameter forces the Intermec FTP Server to set the root of the FTP mount point to the specified value. *Note that this must map to an existing directory or you will not be able to log into the FTP Server.*

Key

HKLM\Software\Intermec\IFTP

Value Type REG_SZ

Valid Range None.

Default

\

Transferring Files Over TCP/IP Networks

The File Transfer Protocol (FTP) server transfers files over TCP/IP networks. The FTPDCE.EXE program is a version that does not display a window, but can run in the background.

FTPDCE is the Internet File Transfer Protocol (FTP) server process. The server can be invoked from an application or command line. Besides servicing FTP client requests, the FTP Server also sends a "network announcement" to notify prospective clients of server availability.

Remarks

The FTP Server currently supports the following FTP requests:

- CDUP Changes to the parent directory of the current working directory.
- CWD Changes working directory.
- DELE Deletes a file.
- HELP Gives help information.
- LIST (*This FTP request is the same as the ls -lgA command*). Gives list files in a directory.
- MKD

Makes a directory.

- MODE (Always Uses Binary). Specifies data transfer mode.
- NLST

Gives a name list of files in directory (this FTP request is the same as the k command).

- NOOP Does nothing.
- PASS Specifies a password.
- **PWD** Prints the current working directory.
- QUIT Terminates session.
- **RETR** Retrieves a file.
- RMD Removes a directory.
- **RNFR** Specifies rename-from file name.

• RNTO

Specifies rename-to file name.

- STOR Stores a file.
- SYST Shows the operating system type of server system.
- **TYPE** (*Binary transfers only.*) Specifies the data transfer type with the Type parameter.
- USER Specifies user name.
- XCUP (*Not Normally Used*) Changes the parent directory of the current working directory.
- XCWD (Not Normally Used) Changes the current directory.
- XMKD (Not Normally Used) Creates a directory.
- XPWD (*Not Normally Used*) Prints the current working directory.
- XRMD (Not Normally Used) Removes a directory.
- SITE

The following nonstandard or operating system (OS)-specific commands are supported by the SITE request. For Microsoft FTP clients, you can send site commands by preceding the command with "quote" such as "quote site status."

• ATTRIB

Gets or sets the attributes of a given file. (SITE ATTRIB)

Usage: QUOTE SITE ATTRIB [+R | -R] [+A | -A] [+S | -S]

[+H | -H] [[path] filename]

- Sets an attribute.
- Clears an attribute.
- *R* Read-only file attribute.
- *A* Archive file attribute.
- *S* System file attribute.
- *H* Hidden file attribute.

To retrieve the attributes of a file, only specify the file. The server response will be: 200-AD SHRCEIX filename

If the flag exists in its position shown above, it is set. Also, in addition to the values defined above, there is also defined:

- *C* Compressed file attribute.
- *E* Encrypted file attribute.
- *I* INROM file attribute.
- X XIP file attribute (execute in ROM, not shadowed in RAM).

• BOOT

Reboots the server OS. This will cause the system on which the server is executing to reboot. The FTP Server will shut down cleanly before reboot. All client connections will be terminated. Cold boot is default except for the PocketPC build in which the default is warm boot.

(SITE BOOT)

Usage: QUOTE SITE BOOT [WARM | COLD]

```
• COPY
```

Copies a file from one location to another. (SITE COPY)

Usage: QUOTE SITE COPY [source] [destination]

Example

```
QUOTE SITE COPY '\Storage Card\one.dat' '\Storage Card\two.dat'
```

• EXIT

Exits the FTP Server. This command will shut down the FTP Server thus terminating all client connections. (SITE EXIT)

Usage: QUOTE SITE EXIT

• HELP

Gives site command help information. (SITE HELP)

Usage: QUOTE SITE HELP [command]

• KILL

Terminates a running program. (SITE KILL)

Usage: QUOTE SITE KILL [program | pid]

- LOG
 - Opens or closes the program log. (SITE LOG)

Usage: QUOTE SITE LOG [open [filename] | close]

• PLIST

Lists the running processes (not supported on all platforms). (SITE PLIST)

Usage: QUOTE SITE PLIST

• RUN

Starts a program running. If the program to run has spaces in path or filename, wrapping the name with single quotes is required.

Usage: QUOTE SITE RUN [program]

Example

QUOTE SITE RUN '\Storage Card\app.exe'

• STATUS

Returns the current settings of the FTP Server. MAC, serial number, model, IP address, network announcement information as well as OS memory usage are returned. (SITE STATUS)

Usage: QUOTE SITE STATUS

• TIMEOUT

Toggles idle timeout between 120 to 1200 seconds (2 to 20 minutes). If this timer expires with no activity between the client and the server, the client connection will be disconnected. If the optional seconds argument is supplied, the server will set the connection timeout to the number of seconds specified. *Default is 120 seconds or 2 minutes.* (SITE TIMEOUT)

Usage: QUOTE SITE TIMEOUT [seconds]

The remaining FTP requests specified in RFC 959 are recognized, but not implemented.

The banner returned in the parenthetical portion of its greeting shows the version number of the FTP Server as well as the MAC address, serial number and OS of the machine hosting the server.

The FTP Server supports browsing from the latest Netscape and Microsoft web browsers. Drag-and-drop capability is available using this environment.

The FTPDCMDS subdirectory contains commands that can be used from the web browser.

- Click EXITME.BIN to execute a SITE EXIT command.
- Click REBOOTME.BIN to execute SITE BOOT command.
- Use the GET command on these files to have the FTP Server execute these commands.
 - Security:

A customer configurable access control list may be installed on the 700 Series Computer. This list will allow customers to restrict access via the FTP Server to the users they wish. This is in addition to the default Intermec account which can be disabled using the -F0 option at runtime.

The access control list is named FTPDCE.TXT and is placed in the same directory on the 700 Series Computer as the FTPDCE.EXE server. The FTP Server will encrypt this file to keep the information safe from unauthorized users. This file is encrypted when the FTP Server is started so a file that is placed onto the 700 Series Computer after the FTP Server starts will require a restart of the FTP Server to take effect.

The format of the FTPDCE.TXT is as follows:

FTPDCE:user1!passwd1<cr><lf>user2!passwd2<cr><lf>user3!passw d3<cr><lf>...



Note: The user accounts and passwords are case sensitive. Once the access control list is encrypted on the 700 Series Computer, the FTP Server will hide this file from users. Once an access control list has been installed on the 700 Series Computer, a new one will not be accepted by the FTP Server until the previous one is removed. Encrypted access control lists are not portable between 700 Series Computers.

Stopping the FTP Server from Your Application

To allow application programmers the ability to programmatically shut down the FTP Server, the FTP Server periodically tests to see if a named event is signaled. The name for this event is "ITC_IFTP_STOP" (no quotes).

For examples on how to use this event, consult the Microsoft Developer Network Library at *http://www.msdn.com*. The MSDN Library is an essential resource for developers using Microsoft tools, products, and technologies. It contains a bounty of technical programming information, including sample code, documentation, technical articles, and reference guides.

Autostart FTP

This automatically starts the FTP Server (FTPDCE.EXE) when the 700 Series Computer is powered on. This is provided with the NDISTRAY program, which displays the popup menu that currently allows you to load and unload the network drivers. Tap the antenna icon in the System Tray of the Today screen *(a sample antenna icon is circled below)* to get this popup menu.



The default is to start the FTP Server at boot time, unless the following registry entry is defined and set to "0" which disables AutoFTP. "1" enables the AutoFTP. The entry can be set from the NDISTRAY pop-up menu by selecting either AutoFTP On or AutoFTP Off.

HKEY LOCAL MACHINE\Software\Intermec\Ndistray\StartupIFTP

These new entries are located below the selections to load the network drivers. If the StartupIFTP registry key is not defined, the FTP Server is loaded by default, to provide "out-of-the-box" capability for customers who want to begin loading files to the 700 Series Computer without any prior configuration.



Note: If a network driver is unloaded using the NDISTRAY popup menu, and the FTP Server is running, the FTP Server is stopped.

On a resume, if AutoFTP is enabled and the FTP Server is running, it is stopped and restarted. NDISTRAY uses a helper application named RESE-TIFTP to implement the restart on resume feature. To do an AutoFTP Installation Check:

- 1 Ensure the FTP Server is running "out-of-the-box" the first time.
- 2 Tap Start → Today to access the Today screen, then tap the antenna icon in the System Tray to bring up the NDISTRAY pop-up menu. Select AutoFTP Off to disable AutoFTP. Do a warm boot and confirm the FTP Server is not running.

• Built-in Ethernet
Wireless 802.11
No networking
AutoFTP On
AutoFTP Off

3 Tap Start → Today to access the Today screen, then tap the antenna icon in the System Tray to bring up the NDISTRAY pop-up menu. Select AutoFTP On to enable AutoFTP, reboot and confirm it is running.



- **4** Unload the network driver when the FTP Server is running and confirm that it is not running any more.
- **5** Load the FTP Server, establish a connection, then suspend and resume. The server should still be running, but the FTP connection to the client should be dropped.

Full Screen

Pocket PC is a hardware specification created by Microsoft Corporation. Devices that wish to carry the Pocket PC logo must meet the minimum hardware requirements set in the Pocket PC specification. Manufacturers are free to add extra hardware functionality.

Pocket PC 2002 devices also use a specialized version of the CE operating system. This OS is built from Windows CE 3.0 but contains customizations, most notably the lack of a desktop and the addition of the Today Screen.

To carry the Pocket PC logo, all devices must be tested at an Independent Test Laboratory. The ITL testing is done based on Microsoft requirements. The test lab then reports the findings back to Microsoft Corporation and Intermec Technologies. If the 700 Series Computer passed all tests, Intermec is allowed to ship the device with the Pocket PC logo. Each time the operating system is modified, Intermec must resubmit to ITL testing.

This means we cannot change the operating system much and still be a Pocket PC device. For example, if we remove Word from the Start menu, the device would fail ITL testing and we would not be able to ship devices with the Pocket PC logo.

Although many customers want a Pocket PC device, some customers would prefer that their users not have access to all of the Pocket PC features. Intermec cannot customize the operating system in any way but a custom application can:

- Delete items from the Start menu, and Programs folder. These items are just shortcuts in the file system so the application is not really being deleted. Cold booting the device will bring these items back so the application will need to be run on every cold boot.
- Use the RegFlushKey() API to save a copy of the registry to a storage device. See the *Recovery CD Help* for more information on how to do this. Saving a copy of the registry will allow most system settings to be restored in a cold boot situation.
- Use the SHFullScreen() API in conjunction with other APIs to make the application take up the entire display and prevent the start menu from being available.
- Remap keys and disable keys on the keypad.
- Create a custom SIP.
- Make changes to the registry to configure the device.

Should you want your 700 Series Computer to display a full screen, keep in mind that your computer is Pocket-PC certified by Microsoft Corporation. Check out resources on programming for the Pocket PC, using the following links. These instructions give full instructions on how to display full screen.

- Instructions on how to create a full screen application for eVC++ applications using an SHFullScreen() API: http://support.microsoft.com/support/kb/articles/Q266/2/44.ASP
- Instructions on how to create a full screen application for eVB applications also using the SHFullScreen() API: http://support.microsoft.com/support/kb/articles/Q265/4/51.ASP

Kernel I/O Controls

This describes the KernelIoControl() functions available to application programmers. Most C++ applications will need to prototype the function as the following to avoid link and compile errors.

extern "C" BOOL KernelIoControl(DWORD dwIoControlCode, LPVOID lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned);

IOCTL_HAL_GET_DEVICE_INFO

This IOCTL returns either the platform type or the OEMPLATFORM name based on an input value.

Syntax

BOOL **KernelIoControl (** IOCTL HAL GET_DEVICE_INFO, LPVOID lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned);

Parameters

lpInBuf	Points to a DWORD containing either the SPI_GETPLATFORMTYPE or SPI_GETOEMINFO value.
lpInBufSize	Must be set to sizeof(DWORD).
lpOutBuf	Must point to a buffer large enough to hold the return data of the function. If SPI_GETPLATFORMTYPE is specified in <i>lpInBuf</i> , then the "PocketPC\0" Unicode string is returned. If SPI_GETOEMINFO is specified in <i>lpInBuf</i> , then the "Intermec 700\0" Unicode string is returned.
nOutBufSize	The size of <i>lpOutBuf</i> in bytes. Must be large enough to hold the string returned.
lpBytesReturned	The actual number of bytes returned by the function for the data requested.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL HAL ITC READ PARM

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelioControl ( IOCTL HAL ITC READ PARM, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

```
Points to this structure. See "ID Field Values" below.
lpInBuf
struct PARMS {
 BYTE id;
 BYTE ClassId;
};
```

nInBufSize	Must be set to the size of the PARMS structure.
lpOutBuf	Must point to a buffer large enough to hold the return data of the function. If this field is set to NULL and <i>nOutBufSize</i> is set to zero when the function is called the function will return the number bytes required by the buffer.
nOutBufSize	The size of <i>lpOutBuf</i> in bytes.
lpBytesReturned	The number of bytes returned by the function for the

data requested.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the error value. Either ERROR_INVALID_PARAMETER or ERROR_INSUFFICIENT_BUFFER may be returned when this function

is used to get the error.

ID Field Values

The *id* field of the PARMS structure may be one of the following values:

- ITC_NVPARM_ETHERNET_ID This IOCTL returns the Ethernet 802.11 MAC Address. Six bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.
- ITC NVPARM SERIAL NUM This IOCTL returns the serial number of the device in BCD format. Six bytes are returned in the buffer pointed to by the lpOutBuffer parameter.
- ITC_NVPARM_MANF_DATE

This IOCTL returns the device date of manufacture in the BCD YYYY/MM/DD format. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

• ITC_NVPARM_SERVICE_DATE

This IOCTL returns the device's date of last service in BCD YYYY/ MM/DD format. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

• ITC_NVPARM_DISPLAY_TYPE

This IOCTL returns the device's display type. One byte is returned in the buffer pointed to by the *lpOutBuffer* parameter.

• ITC_NVPARM_EDG_IP

This IOCTL returns the device Ethernet debug IP address. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

• ITC_NVPARM_EDBG_SUBNET

This IOCTL returns the device Ethernet debug subnet mask. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

• ITC_NVPARM_ECN

This IOCTL returns ECNs applied to the device in a bit array format. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

• ITC_NVPARM_CONTRAST

This IOCTL returns the device default contrast setting. Two bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

• ITC_NVPARM_MCODE

This IOCTL returns the manufacturing configuration code for the device. Sixteen bytes are returned in the buffer pointed to by the *lpOut-Buffer* parameter.

• ITC_NVPARM_VERSION_NUMBER

This IOCTL returns the firmware version for various system components. These values for the *ClassId* field of the PARMS structure are allowed when ITC_NVPARM_VERSION_NUMBER is used in the *id* field:

• VN_CLASS_KBD

Returns a five-byte string, including null terminator, that contains an ASCII value which represents the keyboard microprocessor version in the system. The format of the string is *x.xx* with a terminating null character.

• VN_CLASS_ASIC

Returns a five-byte string, including null terminator, that contains an ASCII value which represents the version of the FPGA firmware in the system. The format of the string is *x.xx* with a terminating null character.

• VN_CLASS_BOOTSTRAP

Returns a five-byte string, including null terminator, that contains an ASCII value which represents the version of the Bootstrap Loader firmware in the system. The format of the string is *x.xx* with a terminating null character.

• ITC_NVPARM_INTERMEC_SOFTWARE_CONTENT

This IOCTL reads the manufacturing flag bits from the non-volatile data store that dictates certain software parameters. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer* that indicates if Intermec Content is enabled in the XIP regions. TRUE indicates that it is enabled. FALSE indicates that it is not enabled.

• ITC_NVPARM_ANTENNA_DIVERSITY

This IOCTL reads the state of the antenna diversity flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer* that indicates if there is a diversity antenna installed. TRUE indicates that it is installed. FALSE indicates that it is not installed.

• ITC_NVPARM_WAN_RI

This IOCTL reads the state of the WAN ring indicator flag. A BOOL-EAN DWORD is returned in the buffer pointed to by *lpOutBuffer* that indicates the polarity of the WAN RI signal. TRUE indicates active high. FALSE indicates active low.

• ITC_NVPARM_RTC_RESTORE

This IOCTL reads the state of the real-time clock restore flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the RTC will be restored upon a cold boot. FALSE indicates that the RTC will not be restored.

• ITC_NVPARM_INTERMEC_DATACOLLECTION_SW

This IOCTL reads the state of the data collection software enabled flag. A BOOLEAN DWORD is returned in the buffer pointer to by *lpOut-Buffer* that indicates the data collection software is to be installed at boot time. FALSE indicates the data collection software should not be installed.

• ITC_NVPARM_INTERMEC_DATACOLLECTION_HW

This IOCTL reads the data collection hardware flags. A BYTE is returned in the buffer pointer to by *lpOutBuffer* that indicates the type of data collection hardware installed. The maximum possible value returned is ITC_DEVID_SCANHW_MAX.

- ITC_DEVID_SCANHW_NONE No scanner hardware is installed.
- ITC_DEVID_OEM2D_IMAGER OEM 2D imager is installed.
- ITC_DEVID_INTERMEC2D_IMAGER Intermec 2D imager is installed.
- ITC_DEVID_SE900_LASER SE900 laser is installed.
- ITC_DEVID_SE900HS_LASER SE900HS laser is installed.

The high bit indicates whether the S6 scanning engine is installed. The bit mask for this is ITC_DEVID_S6ENGINE_MASK. A non-zero value indicates that the S6 scanning engine is installed.

• ITC_NVPARM_WAN_INSTALLED

This IOCTL reads the state of the WAN radio installed flag. A BOOL-EAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the WAN radio is installed. FALSE indicates that no WAN radio is installed.

• ITC_NVPARM_WAN_FREQUENCY

This IOCTL reads the state of the WAN radio frequency flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer.* TRUE indicates that the WAN radio frequency is United States. FALSE indicates that the WAN radio frequency is European.

• ITC_NVPARM_WAN_RADIOTYPE

This IOCTL reads the WAN radio ID installed by manufacturing. A BYTE is returned in the buffer pointer to by *lpOutBuffer* which indicates the type of WAN radio hardware installed. The maximum possible value returned is ITC_DEVID_WANRADIO_MAX. The current definitions are:

- ITC_DEVID_WANRADIO_NONE No WAN radio installed.
- ITC_DEVID_WANRADIO_SIERRA_SB555 CDMA Sierra Wireless radio.
- ITC_DEVID_WANRADIO_XIRCOM_GEM3503 GSM/GPRS Intel (Xircom) radio.
- ITC_DEVID_WANRADIO_SIEMENS_MC45 GSM/GPRS Siemens radio.

• ITC_NVPARM_80211_INSTALLED

This IOCTL reads the state of the 802.11b radio installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer.* TRUE indicates that the 802.11b radio is installed. FALSE indicates that no 802.11b radio is installed.

• ITC_NVPARM_80211_RADIOTYPE

This IOCTL reads the 802.11b radio ID installed by manufacturing. A BYTE is returned in the buffer pointer to by *lpOutBuffer* that indicates the type of 802.11b radio hardware installed. The maximum possible value returned is ITC_DEVID_80211RADIO_MAX. The current definitions are:

- ITC_DEVID_80211RADIO_NONE No 802.11b radio installed.
- ITC_DEVID_80211RADIO_INTEL_2011B Intel 2011B radio installed.

• ITC_NVPARM_BLUETOOTH_INSTALLED This IOCTL reads the state of the Bluetooth radio installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuf-*

fer. TRUE indicates that the Bluetooth radio is installed. FALSE indicates that no Bluetooth radio is installed.

• ITC_NVPARM_SERIAL2_INSTALLED

This IOCTL reads the state of the serial 2 (COM2) device installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the serial 2 device is installed. FALSE indicates that no serial 2 device is installed.

• ITC_NVPARM_VIBRATE_INSTALLED

This IOCTL reads the state of the vibrate device installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuf-fer*. TRUE indicates that the vibrate device is installed. FALSE indicates that no vibrate device is installed.

• ITC_NVPARM_LAN9000_INSTALLED

This IOCTL reads the state of the Ethernet device installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuf-fer*. TRUE indicates that the Ethernet device is installed. FALSE indicates that no Ethernet device is installed.

• ITC_NVPARM_SIM_PROTECT_HW_INSTALLED

This IOCTL reads the state of the SIM card protection hardware installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the SIM card protection hardware is installed. FALSE indicates that no SIM card protection hardware is installed.

• ITC_NVPARM_SIM_PROTECT_SW_INSTALLED

This IOCTL reads the state of the SIM card protection software installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the SIM card protection software is installed. FALSE indicates that no SIM card protection software is installed.

IOCTL_HAL_ITC_WRITE_SYSPARM

Describes and enables the registry save location.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl ( IOCTL_HAL_ITC_WRITE_SYSPARM, LPVOID 
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD 
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	A single byte that may be one of the <i>id</i> values. See " <i>ID Field Values</i> " below.
nInBufSize	Must be set to the size of the <i>lpInBuf</i> in bytes.
lpOutBuf	Must point to a buffer large enough to hold the data to be written to the non-volatile data store.
nOutBufSize	The size of <i>lpOutBuf</i> in bytes.
lpBytesReturned	The number of bytes returned by the function.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the error value. Either ERROR_INVALID_PARAMETER or

ERROR_INSUFFICIENT_BUFFER may be returned when this function is used to get the error.

ID Field Values

The *id* field of *lpInBuf* may be one of the following values:

• ITC_REGISTRY_LOCATION

This IOCTL sets the default location for where to write the registry when RegFlushKey() is called by an application. The registry may be saved to Flash, a CompactFlash storage card or a SecureDigital storage card. *lpOutBuf* must point to a buffer that contains a byte value of "1" for the CompactFlash card or "2" for the SecureDigital card to specify the location.

• ITC_REGISTRY_SAVE_ENABLE

This function enables or disables the save registry to non-volatile media feature of the RegFlushKey() function. *lpOutBuf* must be set to zero (FALSE) if the feature is to be disabled or one (TRUE) if the feature is to be enabled.

• ITC_DOCK_SWITCH

This IOCTL sets a position of the dock switch. The dock switch may be set to either "modem" or "serial" positions. *lpOutBuf* must point to a buffer that contains a byte value of either DOCK_MODEM or DOCK_SERIAL as defined in OEMIOCTL.H; the value specifies the position the switch is to be set.

• ITC_WAKEUP_MASK

This IOCTL sets a bit mask that represents the mask for the five programmable wakeup keys. The I/O key is not a programmable wakeup key. By default it is always the system resume key and all other keys are set to disable key wakeup. A zero in a bit position masks the wakeup for that key. A one in a bit position enables wakeup for that key. *lpOutBuf* must point to a buffer that contains a byte value of a wakeup mask consisting of the OR' ed constants as defined in OEMIOCTL.H. Only the following keys are programmable as wakeup events.

#define SCANNER_TRIGGER1
#define SCANNER_LEFT 2
#define SCANNER_RIGHT 4
#define GOLD_A1 8
#define GOLD_A2 0x10

• ITC_AMBIENT_KEYBOARD

This IOCTL sets the threshold for the keyboard ambient sensor. This can be a value from 0 (always off) to 255 (always on). *lpOutBuf* must point to a buffer that contains a byte value of the desired setting.

• ITC_AMBIENT_FRONTLIGHT

This IOCTL sets the threshold for the frontlight ambient sensor. This can be a value from 0 (always off) to 255. *lpOutBuf* must point to a buffer that contains a byte value of the desired setting.

IOCTL_HAL_GET_DEVICEID

This IOCTL returns the device ID. There are two types of device IDs supported, which are differentiated based on the size of the *output* buffer. The UUID is returned if the buffer size is set to *sizeof(UNIQUE_DEVICEID)*, otherwise the oldstyle device ID is returned.

Usage

#include "pkfuncs.h"
#include "deviceid.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_DEVICEID,LPVOID
lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD
nOutBufSize,LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL. STRICT_ID settings are not supported.
lpInBufSize	Should be set to zero.
lpOutBuf	Must point to a UNIQUE_DEVICEID structure as defined by DEVICEID.H if the UUID is to be returned.
nOutBufSize	The size of the UNIQUE_DEVICEID in bytes if the UUID is to be returned. A DEVICE_ID as defined by PKFUNCS.H is returned if the size in bytes is greater than or equal to <i>sizeof(DEVICE_ID)</i> .

lpBytesReturned The number of bytes returned by the function.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_GET_OAL_VERINFO

Returns the HAL version information of the Pocket PC image.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl ( IOCTL_HAL_GET_OAL_VERINFO, LPVOID 
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Pa

Parameters			
lpInBuf	Should be set to NULL.		
lpInBufSize	Should be set to zero.		
lpOutBuf	Must point to a VERS OEMIOCTL.H. The • cboemverinfo • verinfover • sig; • id; • tgtcustomer • tgtplat • tgtplatversion • tgtcputype[8]; • tgtcpu • tgtcoreversion • date • time	IONINFO structure as defined by fields should have these values: sizeof (tagOemVerInfo); 1 "ITC\0" 'N' "" SeaRay Current build version number "Intel\0" "PXA250\0"; "" Build time Build date	
O D C'	TI : CVEDCION	INIEO 1. 1	

nOutBufSize The size of VERSIONINFO in bytes.

lpBytesReturned Returns *sizeof(PVERSIONINFO)*.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_GET_BOOTLOADER_VERINFO

Returns the HAL version information of the Pocket PC image.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_OAL_VERINFO,LPVOID
lpInBuf, DWORD nInBufSize,LPVOID lpOutBuf,DWORD
nOutBufSize,LPDWORD lpBytesReturned );
```

Parameters

lpInBuf Should be set to NULL.

lpInBufSize Should be set to zero.

lpOutBuf Must point to a VERSIONINFO structure as defined by OEMIOCTL.H. The fields should have these values:

	• cboemverinfo	Sizeof (tagOemVerInfo);
	• verinfover	1
	• sig;	"ITC\0"
	• id;	'В'
	• tgtcustomer	«»
	• tgtplat	SeaRay
	 tgtplatversion 	Current build version number of
		the bootstrap loader
	 tgtcputype[8]; 	"Intel\0";
	• tgtcpu	"PXA250\0"
	• tgtcoreversion	«»
	• date	Build time
	• time	Build date
ıfSize	The size of VERSIO	DNINFO in bytes.

nOutBufSize

lpBytesReturned The number of bytes returned to *lpOutBuf*.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_WARMBOOT

Causes the system to perform a warm-boot. The object store is retained.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl ( IOCTL_HAL_WARMBOOT, LPVOID 
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD 
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBufShould be set to NULL.lpInBufSizeShould be set to zero.lpOutBufShould be NULL.nOutBufSizeShould be zero.

Return Values

None.

IOCTL_HAL_COLDBOOT

Causes the system to perform a cold-boot. The object store is cleared.

Usage

#include "oemioctl.h"

Syntax

BOOL KernelIoControl(IOCTL_HAL_COLDBOOT,LPVOID
lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD
nOutBufSize,LPDWORD lpBytesReturned);

Parameters

lpInBufShould be set to NULL.lpInBufSizeShould be set to zero.lpOutBufShould be NULL.nOutBufSizeShould be zero.

Return Values

None.

IOCTL_HAL_GET_RESET_INFO

This IOCTL code allows software to check the type of the most recent reset.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_RESET_INFO,LPVOID
lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD
nOutBufSize,LPDWORD lpBytesReturned );
```

Parameters

		lpInBuf	Should be	set	to NULL.
		lpInBufSize	Should be	set	to zero.
		lpOutBuf	Must poir	nt to	a HAL_RESET_INFO structure:
typedef DWORD DWORD } HAL_RH	struct { ResetReason; ObjectStoreS ESET_INFO, *	tate; PHAL_RESET_INF	0;	 	most recent reset type state of object store
// Reset	reason type	S			
#define	HAL RESET TY	PE UNKNOWN	0		
#define	HAL RESET RE	ASON HARDWARE	1	11	cold
#define	HAL RESET RE	ASON SOFTWARE	2	//	suspend
#define	HAL RESET RE	ASON WATCHDOG	4		
#define	HAL RESET BA	TT_FAULT	8	//	power fail
#define	HAL_RESET_VD	D_FAULT	16	//	warm boot
		63			

// Object store state flags
#define HAL_OBJECT_STORE_STATE_UNKNOWN 0
#define HAL_OBJECT_STORE_STATE_CLEAR 1

nOutBufSize The size of HAL_RESET_INFO in bytes.

lpBytesReturned The number of bytes returned by the function.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_GET_BOOT_DEVICE

This IOCTL code allows software to check which device CE booted from.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelioControl ( IOCTL HAL GET BOOT DEVICE, LPVOID 
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD 
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Must point to a buffer large enough to hold a DWORD (4 bytes) that contains the boot device. The following boot devices are supported:

#define	HAL BOOT DEVICE UNKNOWN	0
#define	HAL BOOT DEVICE ROM XIP	1
#define	HAL BOOT DEVICE ROM	2
#define	HAL BOOT DEVICE PCMCIA ATA	3
#define	HAL_BOOT_DEVICE_PCMCIA_LINEAR	4
#define	HAL BOOT DEVICE IDE ATA	5
#define	HAL_BOOT_DEVICE_IDE_ATAPI	6

nOutBufSize The size of lpOutBuf in bytes (4).

lpBytesReturned The number of bytes returned by the function.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_REBOOT

Causes the system to perform a warm-boot. The object store is retained.

Usage

#include "oemioctl.h"

Syntax

BOOL **KernelIoControl(** IOCTL HAL REBOOT, LPVOID *lpInBuf*, DWORD *nInBufSize*, LPVOID *lpOutBuf*, DWORD *nOutBufSize*, LPDWORD *lpBytesReturned*);

Parameters

lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Should be NULL.
nOutBufSize	Should be zero.

Return Values None.

IOCTL_PROCESSOR_INFORMATION

Returns processor information.

Usage

#include "pkfuncs.h"

Syntax

```
BOOL KernelIoControl( IOCTL_PROCESSOR_INFORMATION,LPVOID
lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD
nOutBufSize,LPDWORD lpBytesReturned );
```

Parameters

Parameters:	
lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Should be a pointer to the PROCESSOR_INFO structure. The PROCESSOR_INFO structure stores information that describes the CPU more descriptively.

typedef	PROCESSOR INFO {		
WORD	wVersion;	//	Set to value 1
WCHAR	<pre>szProcessorCore[40];</pre>	//	"ARM\0"
WORD	wCoreRevision;	//	4
WCHAR	<pre>szProcessorName[40];</pre>	//	"PXA250\0"
WORD	wProcessorRevision;	//	0
WCAHR	<pre>szCatalogNumber[100];</pre>	//	0
WCHAR	<pre>szVendor[100];</pre>	//	"Intel Corporation\0"
DWORD	dwInstructionSet;	//	0
DWORD	dwClockSpeed;	//	400
}			

nOutBufSize Should be set to sizeof(PROCESSOR_INFO) in bytes.

lpBytesReturned Returns sizeof(PROCESSOR_INFO);

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_GET_CPU_ID

Returns Xscale processor ID.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelloControl( IOCTL_GET_CPU_ID,LPVOID lpInBuf, DWORD
nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD
lpBytesReturned );
```

Parameters

lpInBuf	Should point to a CPUIdInfo structure defined in OEMIOCTL.H.
lpInBufSize	Should be sizeof(CPUIdInfo).
lpOutBuf	Should be NULL.
nOutBufSize	Should be set to 0.
lpBytesReturned	Returns sizeof(PROCESSOR_INFO);

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

Reboot Functions

There are several methods, via Kernel I/O Control functions, that an application program can use to force the 700 Series Computer to reboot.

IOCTL_HAL_REBOOT

IOCTL_HAL_REBOOT performs a warm-boot. See page 278.

IOCTL_HAL_COLDBOOT

Invoking the KernelIOControl function with IOCTL_HAL_COLDBOOT forces a cold reboot. This resets the 700 Series Computer and reloads Windows CE as if a power-up had been performed. The contents of the Windows CE RAM-based object store are discarded. See page 275.

IOCTL_HAL_WARMBOOT

This function is supported on the 700 Series Computers. It performs a warm boot of the system, preserving the object store. See page 275.

Remapping the Keypad

C

Note; Use caution when remapping the keypad. Improper remapping may render the keypad unusable. Data within the 700 Series Computer could also be lost, should any problems occur.

Applications have the ability to remap keys on the 700 Color Keypad. This will allow applications to enable keys that would otherwise not be available, such as the [F1] function key. Also, to disable keys that should not be available, such as the alpha key because no alpha entry is required. Care should be exercised when attempting to remap the keypad because improper remapping may cause the keypad to become unusable. This can be corrected by cold booting the device which will cause the default keymap to be loaded again.

Note that remapping the keys in this way affects the key mapping for the entire system, not just for the application that does the remapping.

There are three "planes" supported for the 740 Keypad. Keys that are to be used in more than one shift plane must be described in each plane.

Unshifted Plane

The unshifted plane contains values from the keypad when not pressed with other keys, such as the following:

- [1] 1
- [5] 5
- [9] 9

Gold Plane

The gold plane contains values from the keypad when a key is simultaneously pressed with the [Gold] key, such as the following:

- [Gold] + [1] Send
- [Gold] + [5] A3
- [Gold] + [9] PageDown

Alpha Plane

The alpha plane contains values from the keypad when the keypad has been placed in alpha mode by pressing the blue alpha key, such as the following:

- [Alpha] + [1] Caps
- [Alpha] + [5] JKL
- [Alpha] + [9] WXYZ

Key Values

Key values for each plane are stored in the registry. All units ship with a default key mapping already loaded in the registry. Applications that wish to change the default mapping need to read the appropriate key from the registry into an array of Words, modify the values required and then write the updated values back into the registry. The registry access can be done with standard Microsoft API calls, such as RegOpenKeyEx(), RegQuery-ValueEx(), and RegSetValueEx().

- The unshifted plane mapping can be found in the registry at: HKEY_LOCAL_MACHI NE\HARDWARE\DEVI CEMAP\KEYBD\Vkey
- The gold plane mapping can be found in the registry at: HKEY_LOCAL_MACHI NE\HARDWARE\DEVI CEMAP\KEYBD\VkeyGol d
- The alpha plane mapping can be found in the registry at: HKEY_LOCAL_MACHI NE\HARDWARE\DEVI CEMAP\KEYBD\VkeyAl pha

How Key Values Are Stored in Registry

To know which fields to update in the registry, you must know what Scan Codes are assigned to each physical key (see the table below). The Scan Code is used at the lowest level of the system to let the keypad driver know which physical key has been pressed. The keypad driver takes that scan code and looks it up in a table (a copy of the one stored in the registry) to determine which values to pass on to the operating system.

Each registry key is just an array that describes to the keypad driver what value needs to be passed for each physical key. The key values are indexed by the scan code, this is a zero-based index. For example in the unshifted plane, the [4] key has a scan code of 0x06. This means that the seventh word under the "Vkey" registry key will have the value for the [4] key. Taking a sample of the "Vkey" registry key shows the following values:

00,00,0B,05,02,03,C1,07,04,03,BE,00,<u>34,00</u>,00,00,. .

The value is 34,00. The values are in reverse byte order because that is the way the processor handles data. When writing an application, nothing needs to be done to swap the bytes, as this will happen automatically when the data is read into a byte value. This is something you just need to be aware of this when looking at the registry. Knowing this, we can see that the value that the keypad driver will pass to the system is a hex 34. Looking that up on an UNICODE character chart, we see that it maps to a "4". If you wanted the key, labeled "4", to output the letter "A" instead, you would need to change the seventh word to "41" (the hexadecimal representation of "A" from the UNICODE chart), then put the key back into the registry.



Note: Do not remap scan codes 0x01, 0x41, 0x42, 0x43, 0x44. Remapping these scan codes could render your 700 Series Computer unusable until a cold-boot is performed.

If you wish to disable a certain key, remap its scan code to 0x00.

Change Notification

Just changing the registry keys will not immediately change the key mappings. To notify the keypad driver that the registry has been updated, signal the "ITC_KEYBOARD_CHANGE" named event using the CreateEvent() API.

Advanced Keypad Remapping

It is also possible to map multiple key presses to one button and to map named system events to a button. The multiple key press option could be useful to cut down on the number of keys needed to press in a given situation or to remap which key behaves like the action key. Mapping events to a button could be useful to change which buttons will fire the scanner, control volume, and allow for suspending and resuming the device. If you need help performing one of these advanced topics please contact Intermec Technical Support.

Scan Codes

At the lowest driver level, the 740 Keypad identifies keys as scan codes. These scan codes are sent via the keypad microcontroller, and cannot be changed without modifying the keypad firmware.

<u>Scancode</u>
0x00
0x01
0x02
0x03
0x04
0x05
0x06
0x07
0x08
0x09
0x0A
0x0B
0x0C
0x0D
0x0E
0x0F
0x10
0x11
0x12
0x13
0x14
0x15
0x16
0x17
0x18
0x19

Key/Meaning	<u>Scancode</u>
Action Key	0x1A
3	0x1B
9	0x1C
ENTER	0x1D
6	0x1E
None	0x1F-0x40
Charge Detect	0x41
LCD Frontlight	0x42
Ambient Light	0x42
Threshold Crossed	0x42
Headset Detected	0x43
Keypad Backlight	0x44
Ambient Light	0x44
Threshold Crossed	0x44

Sample View of Registry Keys

The following is a sample view of the current default key mapping. See the registry on your device for the latest key mappings.

```
[HKEY LOCAL MACHINE\HARDWARE\DEVICEMAP\KEYBD]
"ResumeMask"=dword:7
"Vkey"=hex: 00,00,0B,05,02,03,C1,07,04,03,BE,00,34,00,00,00,
 25,00,00,00,08,00,03,02,00,00,1B,00,28,00,31,00,\
 37,00,01,02,00,00,26,00,27,00,32,00,38,00,30,00,\
 35,00,00,00,01,03,33,00,39,00,0D,00,36,00,00,\
 00,00,07,05,01,05,03,05,02,05
"VkeyGold"=hex: 00,00,0B,05,02,03,C1,07,04,03,BE,00,34,00,00,00,
 09,01,00,00,BF,00,03,02,00,00,BD,00,75,00,72,00,\
 21,00,01,02,00,00,76,00,09,00,73,00,38,01,5B,00,\
 35,00,00,00,BB,01,09,05,22,00,32,01,36,00,00,00,\
 00,00,07,05,01,05,03,05,02,05
"VkeyAlpha"=hex: 00,00,0B,05,02,03,C1,07,04,03,BE,00,47,00,00,00,\
 25,00,00,00,08,00,03,02,00,00,1B,00,28,00,02,02,\
 50,00,01,02,00,00,26,00,27,00,41,00,54,00,20,00,\
 4A,00,00,00,01,03,44,00,57,00,0D,00,4D,00,00,00,\
 00,00,07,05,01,05,03,05,02,05
```



This appendix contains information about the Data Collection, SNMP, and User Information Control Panel applets that may be on your 700 Series Color Mobile Computer.

SNMP and Data Collection settings that can appear under **Settings** are dependent on what hardware configuration is done for each 700 Series Computer at the time of shipment. These settings will currently only appear if a scanner or an imager option is present.

Likewise, other control panel applets that are specifically related to the 802.11b radio module will appear when a 802.11b radio module is installed in a 700 Series Computer. Control panel applets that are specific for Wireless Printing, CDMA/1xRTT, and GSM/GPRS radio modules will only appear when each respective hardware configuration is done on the 700 Series Computer. See Chapter 4, "Network Support," for more information about the radio modules or the wireless printing.

Configuration Parameters

A configuration parameter changes the way the 700 Series Color (700C) Mobile Computer operates, such as configuring a parameter to have the 700 Series Computer emit a very loud beep in a noisy environment. Use any of the following methods to execute configuration parameters:

- Change Data Collection and SNMP parameters via control panel applets later in this appendix.
- Access the 700 Series Computer via the Unit Manager through a web browser on your desktop PC via the SRDEVMGMT.CAB file. To use the Unit Manager, install this CAB file from the 700 Color Software Tools CD-ROM. Unit Manager applications are available on the 700 Series Color Unit Manager CD-ROM. For more information, consult your Intermec sales representative.
- Send parameters from an SNMP management station. See "SNMP Configuration" starting on page 123.
- Scan EasySet bar codes. You can use the EasySet bar code creation software from Intermec Technologies Corporation to print configuration labels. Scan the labels to change the scanner configuration and data transfer settings.

Changing a Parameter Setting

Menus of available parameters for each group are listed. Use the scroll bars to go through the list. Expand each menu (+) to view its parameter settings. Tap a parameter to select, or expand a parameter (+) to view its subparameters.

Note that each parameter or subparameter is shown with its default setting or current setting in (< >) brackets. Tap a parameter or subparameter to select that parameter, then do any of the following to change its setting: Tap **Apply** to apply any changes. *Note that these illustrations are from a Symbologies parameter*.

- Typing a new value in an entry field.
- Choosing a new value from the drop-down list.
- Selecting a different option. The selected option contains a bullet.
- Tap **Defaults**, then **Apply** to restore factory-default settings. Tap **Yes** when you are prompted to verify this action.

SYMBOLOGIES				
Caution restoring Factory Defaults will replace all settings for this page. Restore Factory Defaults now?				
Yes No				

• Tap **Refresh** to discard changes and start again. Tap **Yes** when you are prompted to verify this action.

SYMBOLOGIES			
Refresh will discard all unapplied edits and refetch values for this page. Refresh now ?			
Yes No			

About Configuration Parameters

You can find the following information about each configuration parameter:

- Name and Purpose: Describes the parameter and its function.
- Action: Describes what to do with a parameter once that parameter is selected.
- **SNMP OID**: Lists the SNMP OID for the parameter.
- Syntax or Options:

Syntax lists the two-character code for the parameter, if the parameter is configurable by scanning a bar code or by sending parameters through a network. Both **Syntax** and **Options** list acceptable values for the parameter. *Default settings are noted in italic.*

Data Collection Control Panel Applet

See "Scanner Control and Data Transfer" in the Intermec Windows CE/Pocket PC Software Developer's Kit (SDK) User's Manual shipped with the Software Developer's Kit (SDK) for information about data collection functions.



Note: Icons are shown to the left.

To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Data Collection icon to access its control panel applet.

8 Setting	5	🧲 1:32 🛛 🔂
Data Collec	tion	
 Gode 39, Standard Codabar, UPC/EAN Code 93, 	<pre><100100006> 2 of 5, <000100600 , <000010060000000 l, <11110001111110 <not active=""></not></pre>	00000000> = 000> 00>
	III	
Defaults	<u>R</u> efresh	Apply
Symbologies	Symbology Options	Beeper 4
		₩

Use the left and right arrows to scroll through the tabs along the bottom of the control panel applet, then tap a tab to access its menus. These tabs represent the following groups of settings or parameters:

- Symbologies
- Symbology Options (starting on page 309)
- Beeper/LED (starting on page 317)
- Imager (starting on page 323)
- Virtual Wedge (starting on page 325)
Symbologies

You can change bar code symbology parameter settings in your 700 Series Computer via the **Data Collection** control panel applet. The following parameters are for bar code symbologies. Additional information about the more common bar code symbologies are in Appendix C, "*Bar Code Symbologies.*" *Note that these parameters are listed in the order of their appearance within this tab.*

Most of these symbologies apply to both the imager and the laser scanner tools. However, when using an imager, the Macro PDF (*page 300*), Micro PDF 417 (*page 302*), Matrix 2 of 5 (*page 304*), Telepen (*page 305*), and Code 11 (*page 306*) symbologies are not supported. Likewise, when using a laser scanner, the QR Code (*page 307*) and Data Matrix (*page 308*) symbologies are not supported.

The following table shows which bar code symbologies are supported either by an imager or by a laser scanner.

Bar Code Symbology	Imager	Laser Scanner
Code 39	Х	Х
Interleaved 2 of 5	Х	Х
Standard 2 of 5	Х	Х
Matrix 2 of 5		Х
Code 128	Х	Х
Code 93	Х	Х
Codabar	Х	Х
MSI		Х
Plessey		Х
UPC	Х	Х
EAN/EAN 128	Х	Х
Code 11		Х
PDF 417	Х	Х
Micro PDF 417		Х
Telepen		Х
Data Matrix	Х	
QR Code	Х	

Code 39

Code 39 is a discrete, self-checking, variable length symbology. The character set is uppercase A-Z, 0-9, dollar sign (\$), period (.), slash (/), percent (%), space (), plus (+), and minus (-).

Action

Tap (+) to expand the Code 39 parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.3.1

^		. •			
"	n		^	n	c
U	IJ	u	U		з
-	г	•••	-	•••	-

Decoding	0	Not active
Ũ	1	Active (default)
Format	0	Standard 43 characters (default)
	1	Full ASCII
Start/Stop	0	Not transmitted (default)
-	1	Transmitted
Start/Stop characters (Not support	ed when using an imager):
-	0	\$ (dollar sign) only
	1	* (asterisk) only (default)
	2	& and * (dollar sign and asterisk)
Check digit	0	Not used (default)
C	1	Mod 43 transmitted
	2	Mod 43 not transmitted
	3	French CIP transmitted
	4	French CIP not transmitted
	5	Italian CPI transmitted
	6	Italian CPI not transmitted
Bar code length	0	Any length <i>(default)</i>
C	1	Minimum length
Minimum length	000-254	Minimum length 1-254 (6)
Note: If Bar code len	gth = "1" th	en Minimum length is entered.



	rcings			12:05	UK
Data C	ollecti	ion			
🖻 Cod	e 39, <	:100100006:	>		
	Decodi	ng, <active:< td=""><td>></td><td></td><td>=</td></active:<>	>		=
	Format	, <standard< td=""><td>43 cha</td><td>racters></td><td>·П</td></standard<>	43 cha	racters>	·П
	Start/S	itop, <not td="" tr<=""><td>ansmitt</td><td>ed></td><td></td></not>	ansmitt	ed>	
-	Start/S	top characte	ers, <*	only>	-
4	• •		•		
Default	s	<u>R</u> efresh]	App	y

Standard 2 of 5

Standard 2 of 5 is a discrete and self-checking symbology that uses the bars to encode information and the spaces to separate the individual bars.

Action

Tap (+) to expand the **Standard 2 of 5** parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.4.1

Options

Decoding	0	Not active (<i>default</i>)
C	1	Active
Format	0	Identicon, 6 start/stop bars (default)
	1	Computer Identics, 4 start/stop
Check digit	0	Not used (default)
-	1	Mod 10 transmitted
	2	Mod 10 not transmitted
Bar code length	0	Any length
-	1	Minimum length (default)
	2	Fixed lengths
Minimum length	001-254	Minimum length 1-254 (6)
Fixed length 1	000-254	Fixed bar code length 0-254 (0)
Fixed length 2	000-254	Fixed bar code length 0-254 (0)
Fixed length 3	000-254	Fixed bar code length 0-254 (0)



Note: If Bar code length = "1" then Minimum length is entered. If Bar code length = "2" then Fixed length 1, Fixed length 2, or Fixed length 3 is entered.

🔠 Setting	s (24	(11:44 🛛 🚯
Data Collec	tion	
- Standard Deco Form Chec Bar c	2 of 5, <100100600 ding, <active> at, <identicon, 6="" sta<br="">k digit, <not used=""> ode length, <minimur< td=""><td>n length > 🗸</td></minimur<></not></identicon,></active>	n length > 🗸
	 W	•
Defaults	<u>R</u> efresh	Apply
Symbologies	Symbology Options	Beeper 🖣 🕨
		×

Codabar

Codabar is a self-checking, discrete symbology.

Action

Tap (+) to expand the **Codabar** parameter, select a setting to be changed, then select an option from the drop-down list to change this setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.5.1

Options

Decoding	0	Not active (<i>default</i>)
C	1	Active
Start/Stop	0	Not transmitted (default)
-	1	abcd transmitted
	2	ABCD transmitted
	3	abcd/tn*e transmitted
	4	DC1`DC4 transmitted
CLSI library system (Not	supported wh	ben using an imager):
	0	Not active (default)
	1	Active
Check digit	0	Not used (<i>default</i>)
C C	1	Transmitted
	2	Not transmitted
Bar code length	0	Any length
-	1	Minimum length (default)
	2	Fixed lengths
Minimum length	003-254	Minimum length 3-254 (6)
Fixed length 1	000-254	Fixed length 0-254 (0)
Fixed length 2	000-254	Fixed length 0-254 (0)
Fixed length 3	000-254	Fixed length 0-254 (0)



Note: If Bar code length = "1" then Minimum length is entered. If Bar code length = "2" then Fixed length 1, Fixed length 2, or Fixed length 3 is entered.

🔠 Settings	🔁 📢 12:11 🛛 🚯
Data Collection	
Codabar, <100 Decoding, Start/Stop, CLSI library Check digit	0010060000000> <active> = , <not transmitted=""> > y system, <not active=""> ., <not used=""></not></not></not></active>
 Defaults	Refresh Apply
Symbologies Sym	bology Options Beeper
	· [*

UPC/EAN

UPC/EAN are fixed-length, numeric, continuous symbologies that use four element widths.

Action

Tap (+) to expand the UPC/EAN parameter, select the setting to be changed, then select an option to change this setting.

SNMP OID

1 3 6 1 / 1 1063 15 3 3 1 1 6 1

1.3.6.1.4.1.1963.15.3.3.1.	1.6.1
Options	
UPC A	0 Not active
	1 Active (default)
UPC E	0 Not active
	1 Active (default)
EAN 8	0 Not active
	1 Active (default)
EAN 13	0 Not active
	1 Active (default)
Add-on digits	0 Not required (default)
C	1 Required
Add-on 2 digits	0 Not active (default)
-	1 Active
Add-on 5 digits (Not suppo	orted when using an imager):
	0 Not active (<i>default</i>)
	1 Active
UPC A check digit	0 Not transmitted
	1 Transmitted (default)
UPC E check digit	0 Not transmitted
	1 Transmitted (<i>default</i>)
EAN 8 check digit	0 Not transmitted
	1 Transmitted (<i>default</i>)
EAN 13 check digit	0 Not transmitted
	1 Transmitted (default)
UPC A number system	0 Not transmitted
	1 Transmitted (default)
UPC E number system	0 Not transmitted
	1 Transmitted (<i>default</i>)
UPC A re-encoding	0 UPC A transmitted as EAN 13 (default)
	1 UPC A transmitted as UPC A
UPC E re-encoding	0 UPC E transmitted as UPC E (default)
	1 UPC E transmitted as UPC A
EAN 8 re-encoding	0 EAN 8 transmitted as EAN 8 (default)
	1 EAN 8 transmitted as EAN 13



Code 93

Code 93 is a variable length, continuous symbology that uses four element widths.

Action

Tap the **Code 93** parameter, then select an option to change this parameter setting. Tap (+) to access the **Code 93 Lengths** parameter.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.7.1

Options

0 Not active (*default*)

1 Active

Code 93 Length

Sets the Code 93 bar code length.

Action

Tap (+) to expand the **Code 93** parameter, then tap (+) to expand the **Code 93 Lengths** parameter. Tap the setting to be changed, then tap an option to change this setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.19.1

Options

Bar code length	0	Any length (default)
C	1	Minimum length
Minimum length	001-254	Minimum length 1-254 (6)



Minimum length is entered.

街 Settings	4 € 12:10 🚯
Data Collection	
Code 93, <not activ<br="">Code 93 length, Bar code leng </not>	>> <0006> gth, <any length=""> = th, <6></any>
Defaults Refre	sh <u>Apply</u>
Symbologies Symbology	Options Beeper
	E

Code 128

Code 128 is a variable-length, continuous, high-density, alphanumeric symbology that uses multiple element widths and supports the extended ASCII character set.

Action

Tap the **Code 128** parameter, then select an option to change this parameter setting. *The following illustration is for a 700 Series Computer using a laser scanner.*

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.9.1

Options

0 Not active (*default*)

1 Active

街 Settings 🛛 ┥	÷ 12:19 🐽
Data Collection	
Code 128, <not active=""> ⊕ Code 128 options, <10000 Code 128 ENC1 character</not>)6>
⊕ · Plessey, <000006> ⊕ · MSI, <001006>	
Change 'Code 128' Setting —— Not active	
O Active	
Defaults Refresh	Apply
Symbologies Symbology Options	Beeper
	⊠ ^

Code 128 Options

Set the following for the Code 128 parameter. Note that the EAN 128]C1 and CIP 128 French Pharmaceutical options are not available when you use an imager with your 700 Series Computer.

Action

Tap (+) to expand the **Code 128 Options** parameter, select a setting, then select an option to change this setting.

SNMP OID

None.

Options

EAN 128]C1 Identi	fier (disablea	l when using an imager)
	0	Remove
	1	Include <i>(default)</i>
CIP 128 French Pha	rmaceutical	(disabled when using an imager)
	0	Not active (<i>default</i>)
	1	Active
Bar code length	0	Any length <i>(default)</i>
-	1	Minimum length
Minimum length	001-254	Minimum length 1-254 (6)

街 Settings	📢 12:12 🛛 🚯	
Data Collection		
⊡- Code 128 options, <100006> — EAN 128]C1 Identifier, <include — CIP 128 French Pharmaceutical, — Bar code length, <any length=""> Minimum Length <</any></include 		
Defaults Refresh		
Symbologies Symbology C	ptions Beeper 4	
		

Code 128 FNC1 Character

The Code 128 FNC1 character (EAN 128 norms) can be any ASCII character and is used as a separator when multiple identifiers and their fields are concatenated. *Note that this is not available when you use an imager with your 700 Series Computer.*

Non-printable ASCII characters can be entered using the following syntax where *HH* is the hexadecimal value of the character.

\xHH

For example, the GS character, whose hexadecimal value is 1D, would be entered as $\1D$. In addition, the following characters have their own identifiers:

- BEL \a
- BS \b
- FF \f
- LF \n
- CR \r
- HT \t
- VT \v

Action

Tap (+) to expand the **Code 128** parameter, then type the ASCII characters to be set for the **Code 128 FNC1 character** parameter.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.21.1

Options

Any ASCII character (default is the GS function character - ID hex)

<i>B</i> Settings	📢 12:13 🛛 🐟	
Data Collection		
⊖- Code 128, <not active:<="" td=""><th>> _</th></not>	> _	
E • Code 128 options, <	<100006>	
⊕ MSI, <001006>	~	
Change 'Code 128 FNC1 character' Setting		
\x1D		
Defaults Refrest	Apply	
Symbologies Symbology C)ptions Beeper	
	₩	

Plessey

Plessey is a pulse-width modulated symbology like most other bar codes. It includes a start character, data characters, an eight-bit cyclic check digit, and a termination bar. The code is continuous and not self-checking. You need to configure two parameters for Plessey code: Start Code and Check Digit. *Note that this is not available when you use an imager with your 700 Series Computer.*

Action

Tap (+) to expand the **Plessey** parameter, select the setting to be changed, then select an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.10.1

Options

Decoding	0	Not active <i>(default)</i>
U	1	Active
Check digit	0	Not transmitted (default)
-	1	Transmitted
Bar code length	0	Any length <i>(default)</i>
-	1	Minimum length
Minimum length	001-254	Minimum bar code length 1-254 (6)



Note: If Bar code length = "1" then Minimum length is entered.

🗐 Settings	4 € 12:15 🕕
Data Collection	
⊡ Plessey, <000006	> tactive> vot transmitted> tansmitted> ta
Defaults Re	fresh <u>A</u> pply
Symbologies Symbol	ogy Options Beeper
	▲

MSI

MSI is a symbology similar to Plessey code (page 298) that includes a start pattern, data characters, one or two check digits, and a stop pattern. *Note that this is not available when you use an imager with your 700 Series Computer.*

Action

Tap (+) to expand the MSI parameter, select the setting to be changed, then select an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.15.1

Options

0	Not active <i>(default)</i>
1	Active
0	Mod 10 transmitted (default)
1	Mod 10 Not transmitted
2	Double Mod 10 transmitted
3	Double Mod 10 not transmitted
0	Any length
1	Minimum length (default)
001-254	Minimum length 1-254 (6)
	0 1 0 1 2 3 0 1 001-254

P

Note: If Bar code length = "1" then Minimum length is entered.

🕮 Setting	s 🖣	é 12:15 🛛 🔂
Data Collec	tion	
⊡- MSI, <001 Decod Check Bar co Minimu	.006> ing, <not active=""> digit, <mod 10="" trans<br="">de length, <minimum im length, <6></minimum </mod></not>	mitted>
	Ш	
<u>D</u> efaults	<u>R</u> efresh	Apply
Symbologies	Symbology Options	Beeper 4 🕨
	-	- E

PDF 417

PDF 417 is a stacked two-dimensional symbology that provides the ability to scan across rows of code. Each row consists of start/stop characters, row identifiers, and symbol characters, which consist of four bars and four spaces each and contain the actual data. This symbology uses error correction symbol characters appended at the end to recover loss of data.

Because the virtual wedge translates incoming data into keypad input, the size of the keypad buffer limits the effective length of the label to 128 characters. Longer labels may be truncated. For PDF 417 labels of more than 128 characters, you can develop an application that bypasses the keypad buffer.

Action

Tap the PDF 417 parameter, then select an option to change this parameter setting. Tap (+) to access either the Macro PDF options parameter or the Micro PDF 417 parameter.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.17.1

Options

- 0 Not active
- 1 Active (*default*)

🔠 Settings	📢 12:19 🐽
Data Collection	,
PDF 417, <activ acro="" acrow="" activ="" arrow="" o="" pdf="" pdf-4:="" pdf-4:<="" td=""><th>e></th></activ>	e>
Change 'PDF 417' : Not act	ive
Defaults	Refresh Apply
Symbologies Symb	ology Options Beeper

Macro PDF options

Macro PDF is used when a long message requires more than one PDF 417 label. *Note that this is not available when you use an imager with your 700 Series Computer.*

• Select **Buffered** to store a multi-label PDF 417 message in the Sabre buffer, thus transmitting the entire message when all labels have been read.

• Select Unbuffered for multi-label PDF 417 messages that are too long for the Sabre buffer (memory overflow). Each part of the PDF 417 label is transmitted separately, and the host application must then assemble the message using the macro PDF control header transmitted with each label. *Control Header is only present in macro PDF codes and is always transmitted with unbuffered option.*

Action

Tap (+) to expand the PDF 417 parameter, tap (+) to expand the Macro PDF parameter, select a setting to be changed, then select an option to change this setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.22.1

Options

• p • •	
Macro PDF	0 Unbuffered
	1 Buffered (<i>default</i>)
Control header	0 Not transmitted (default)
	1 Transmitted
File name	0 Not transmitted (<i>default</i>)
	1 Transmitted
Segment count	0 Not transmitted (<i>default</i>)
0	1 Transmitted
Time stamp	0 Not transmitted (<i>default</i>)
-	1 Transmitted
Sender	0 Not transmitted (<i>default</i>)
	1 Transmitted
Addressee	0 Not transmitted (<i>default</i>)
	1 Transmitted
File size	0 Not transmitted (<i>default</i>)
	1 Transmitted
Checksum	0 Not transmitted (<i>default</i>)
	1 Transmitted



Micro PDF 417

Micro PDF 417 is a multi-row symbology derived from and closely based on PDF 417 (*page 300*). A limited set of symbology sizes is available, together with a fixed level of error correction for each symbology size. *Note that this is not available when you use an imager with your 700 Series Computer.*

Action

Tap (+) to expand the PDF 417 parameter, tap (+) to expand the Micro PDF 417 parameter, select a setting to be changed, then select an option to change this setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.27.1

Options

Decoding

- 0 Not active (*default*)
- 1 Active 0 Not ac

1

Code 128 Emulation

Not active *(default)* Active

🕖 Settings	📢 12:22 🕕
Data Collection	
)> 🔺
Decoding, <no< td=""><td>t active></td></no<>	t active>
Code 128 Emul	ation, <not active<="" td=""></not>
i∰- Interleaved 2 of 5, <00)100600000000> ≡
⊕- Matrix 2 of 5, <01006>	· 🖌
Defaults Refres	h <u>Apply</u>
Symbologies Symbology (Options Beeper
	₩

Interleaved 2 of 5

Interleaved 2 of 5 (I 2 of 5) is a high-density, self-checking, continuous, numeric symbology used mainly in inventory distribution and the automobile industry.



Note: An Interleaved 2 of 5 bar code label must be at least three characters long for the 700 Series Computer to scan and decode correctly.

Action

Tap (+) to expand the **Interleaved 2 of 5** parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.23.1

Options

Decoding	0	Not active (<i>default</i>)
-	1	Active
Check digit	0	Not used <i>(default)</i>
U	1	Mod 10 transmitted
	2	Mod 10 not transmitted
	3	French CIP transmitted
	4	French CIP not transmitted
Bar code length	0	Any length
-	1	Minimum length (default)
	2	Fixed lengths
Minimum length	003-254	Minimum length 3-254 (6)
Fixed length 1	003-254	Fixed length 3-254 (0)
Fixed length 2	003-254	Fixed length 3-254 (0)
Fixed length 3	003-254	Fixed length 3-254 (0)



Note: If Bar code length = "1" then Minimum length is entered. If Bar code length = "2" then Fixed length 1, Fixed length 2, or Fixed length 3 is entered.

🔠 Settings 🛛 📢	12:22	6	k
Data Collection			
Interleaved 2 of 5, <00100600 Orecoding, <not active=""> Orecoding, <not used=""> Orecode length, <minimum <6="" length,="" orecode=""> III</minimum></not></not>	length>		
Defaults Refresh	App	ly	
Symbologies Symbology Options	Beeper	∢	Þ
	E	8	*

Matrix 2 of 5

Matrix 2 of 5 is a numerical symbology. *Note that this is not available when you use an imager with your 700 Series Computer.*

Action

Tap (+) to expand the Matrix 2 of 5 parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.24.1

Options

Decoding	0	Not active (<i>default</i>)
-	1	Active
Bar code length	0	Any length
-	1	Minimum length (default)
Minimum length	001-254	Minimum length 1-254 (6)



Note: If Bar code length = "1" then Minimum length is entered.

街 Settings	📢 12:24 🛛 🚯
Data Collection	
Matrix 2 of 5, <01006> Decoding, <not act<br="">Bar code length, <p< td=""><td>ive> /inimum length></td></p<></not>	ive> /inimum length>
Minimum length, <6	> 🗐
Telepen, <00>	•
Defaults Refresh	Apply
Symbologies Symbology O	ptions Beeper 🖣 🕨
	₩

Telepen

Telepen is an alphanumeric, case-sensitive, full ASCII symbology. *Note that this is not available when you use an imager with your 700 Series Computer.*

Action

Tap (+) to expand the **Telepen** parameter, select the setting to be changed, then tap an option to change this setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.25.1

Options

Decoding 0 Not active (default) 1 Active Format 0 ASCII (default) 1 Numeric

街 Settings	📢 12:25 🛛 🚯
Data Collection	
∰ Matrix 2 of 5, <01006>	
□ Telepen, <00>	
Decoding, <not active<="" td=""><td>≥ </td></not>	≥
	
Defaults Refresh	
Symbologies Symbology Opti	ions Beeper
	E

Code 11

Code 11 is a high density, discrete numeric symbology that is extensively used in labeling telecommunications components and equipment. *Note that this is not available when you use an imager with your 700 Series Computer.*

Action

Tap (+) to expand the Code 11 parameter, select the setting to be changed, then tap an option to change this setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.26.1

Options

Decoding	0
-	1
Check digit verification	1
e	2
Check digit transmit	0
U U	1

Not active *(default)* Active 1 digit *(default)* 2 digits Disable *(default)* Enable

🔠 Setting	is 📢	(12:26 🐽
Data Collec	tion:	
🗄 - Telepen,	<00>	
⊡ Code 11,	<011>	
Deco	ding, <not active=""></not>	
Chec	k digit verification, <:	ldigit> ⊨
Chec	K digit transmit, <ena< td=""><td>able></td></ena<>	able>
	III	
Defaults	Refresh	Apply
Symbologies	Symbology Options	Beeper

QR Code

QR Code (Quick Response Code) is a two-dimensional matrix symbology containing dark and light square data modules. It has position detection patterns on three of its four corners and features direct encodation of the Japanese Kana-Kanji character set. It can encode up to 2509 numeric or 1520 alphanumeric characters and offers three levels of error detection. *Note that this is not available when you use a laser scanner with your 700 Series Computer.*

Action

Tap (+) to expand the **QR Code** parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.35.1

Options

Decoding 0 Not active 1 Active (*default*)

🔠 Settings		4 € 11:49 😡
Data Collecti	on	
B Data Matrix Change 'Decoo ● No	, <1> III Jing' Setting —— t active	▼
	tive	
Defaults	<u>R</u> efresh	Apply
Symbologies S	iymbology Option	ns Beeper
New Tools Se	rvices 📙 🛃	

Data Matrix

A two-dimensional matrix symbology, which is made of square modules arranged within a perimeter finder pattern. The symbology utilizes Error Checking and Correcting (ECC) algorithm with selectable levels for data error recovery and Cyclic Redundancy Check algorithm to validate the data. The character set includes either 128 characters conforming to ISO 646 (ANSI X3.4 - 1986) or 256 extended character set. Maximum capacity of a symbol is 2335 alphanumeric characters, 1556 8-bit byte characters or 3116 numeric digits. *Note that this is not available when you use a laser scanner with your 700 Series Computer.*

Action

Tap (+) to expand the **Data Matrix** parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.34.1

Options

Decoding 0 Not active 1 Active (*default*)

Settings		€ € 11:50 O B
Data Collection		
- PDF 417, <acti< td=""><td>ve></td><td></td></acti<>	ve>	
₿•Interleaved 2 o	f 5, <001006	0000000000
Br-QR Code, <0>		
🗄 Data Matrix, <1	1>	=
Decoding, <	Active>	-
•		
-Change 'Decoding'	' Settina ——	
🗌 🗌 🔿 Not ac	tive	
Active		
<u>D</u> efaults	<u>R</u> efresh	Apply
Symbologies Symb	bology Option	ns Beeper 🖣 🕨

Symbology Options



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Data Collection icon to access its control panel applet.

Use the right and left arrows to scroll to the **Symbology Options** tab, then tap this tab to access its parameters. The following are parameters for bar code symbology options. *Note that these are listed in the order of their appearance within the Symbology Options tab.*

Symbology ID

Identifies the bar code symbology in which data has been encoded by prepending a user-specified symbology identifier to the data. You can prepend one of these types of character strings to identify the symbology:

• User-defined ASCII Character (Option 1):

A user-defined symbology identifier is a single ASCII character. You can assign a custom identifier character to each bar code symbology. *Note that this is not available when you use an imager with your 700 Series Computer.*

• AIM ISO/IEC Standard (*Option 2 - Required to define symbology IDs*): The AIM Standard has a three-character structure which indicates the symbology and optional features. See the *AIM ISO/IEC Standard* for more information.

Action

Select **Symbology ID**, then select an option to change this parameter setting. Tap (+) to expand the **Symbology ID** parameter, then select any of the user ID parameters listed. *See the top of the next page for a sample screen of the Code 39 user ID*.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.4.1.22.1

Options

- 0 Disable (*default*)
- 1 User defined (disabled when using an imager)
- 2 ISO/IEC Standard

🕀 Settings	4 € 12:40 @
Data Collection	
Symbology ID, <disables 39="" <*="" code="" id,="" user=""> Code 128 user ID, <d: 128="" 2="" 39="" 5="" <d:="" code="" id,="" interleaved="" of="" th="" user="" user<=""><th> ▲ ■ ✓ ID, <i></i> </th></d:></disables>	 ▲ ■ ✓ ID, <i></i>
Change Symbology ID' Setting Disable User defined ISO/IEC Standard]
Defaults Refresh	
Symbology Options Beeper/LE	ED/Buttons

🔠 Settings	📢 3:16 🛛 🐽
Data Collection	
Change 'Code 39 user ID, Code 39 user ID, Code 128 user ID, Code 128 user ID, Code 93 user ID, Code 93 user ID, Code 93 user ID,	≥> ▲ <d> D> D> Jser ID, <i> ▼ 5etting</i></d>
L Defaults Refresh	Apply
Symbologies Symbology O	ptions Beeper
	^

Code 39 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 39 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Code 39 user ID parameter, then enter a user ID value to change this parameter setting.
- SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.3.1
 - **Options:** x where x is a single ASCII character. *Default is asterisk* (*).

Code 128 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 128 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Code 128 user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.5.1
 - **Options:** x where x is a single ASCII character. *Default is asterisk* (*).

Codabar User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Codabar bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Codabar user ID parameter, then enter a user ID value to change this parameter setting.
- SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.2.1

Options: *x* where *x* is a single ASCII character. *Default is D*.

Code 93 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 93 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Code 93 user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.4.1
 - **Options:** x where x is a single ASCII character. *Default is asterisk* (*).

Interleaved 2 of 5 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Interleaved 2 of 5 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Interleaved 2 of 5 user ID parameter, then enter a user ID value to change this parameter setting.
- SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.10.1
 - **Options:** *x* where *x* is a single ASCII character. *Default is I (not lowercase L).*

PDF-417 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify PDF 417 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the PDF 417 user ID parameter, then enter a user ID value to change this parameter setting.
- SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.12.1
 - **Options:** x where x is a single ASCII character. *Default is an asterisk* (*).

MSI User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify MSI bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the MSI user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.11.1
 - **Options:** *x* where *x* is a single ASCII character. *Default is D*.

Plessey User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Plessey bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Plessey user ID parameter, then enter a user ID value to change this parameter setting.
- SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.13.1

Options: *x* where *x* is a single ASCII character. *Default is D*.

Standard 2 of 5 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Standard 2 of 5 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Standard 2 of 5 user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.23.1
 - **Options:** *x* where *x* is a single ASCII character. *Default is D*.

UPC A User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify UPC-A (Universal Product Code) bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the UPC A user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.6.1
 - **Options:** *x* where *x* is a single ASCII character. *Default is A*.

UPC E User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify UPC-E bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the UPC E user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.7.1
 - **Options:** *x* where *x* is a single ASCII character. *Default is E.*

EAN 8 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify EAN-8 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the EAN 8 user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.8.1
 - **Options:** x where x is a single ASCII character. *Default is* xFF.

EAN 13 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify EAN-13 (European Article Numbering) bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the EAN 13 user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.9.1
 - **Options:** *x* where *x* is a single ASCII character. *Default is F.*

Matrix 2 of 5 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Matrix 2 of 5 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Matrix 2 of 5 user ID parameter, then enter a user ID value to change this parameter setting.
- SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.24.1
 - **Options:** *x* where *x* is a single ASCII character. *Default is D*.

Telepen User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Telepen bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

- Action: Tap (+) to expand the Symbology ID parameter, select the Telepen user ID parameter, then enter a user ID value to change this parameter setting.
- **SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.25.1
 - **Options:** x where x is a single ASCII character. *Default is an asterisk* (*).

Code 11 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 11 bar code data. *Note that this is not available when you use an imager with your 700 Series Computer.*

Action: Tap (+) to expand the Symbology ID parameter, select the Code 11 user ID parameter, then enter a user ID value to change this parameter setting.

SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.16.1

Options: x where x is a single ASCII character. *Default is asterisk* (*).

Prefix

Prepends a string of up to 20 ASCII characters to all scanned data.

Action

Tap the **Prefix** parameter, then enter a prefix value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.4.1.29.1

Options

Acceptable values are up to 20 ASCII characters. Embedded null (<NUL >) characters are not allowed. *Default is no characters (disabled).*

街 Settings	🕂 12:37 🕕
Data Collection	
Symbology ID, <disable> Prefix, <> Suffix, <></disable>	
Change 'Prefix' Setting	
Defaults Refresh	Apply
Symbologies Symbology Optic	ons Beeper
	· 🖂

Suffix

Appends a string of up to 20 ASCII characters to all scanned data.

Action

Tap the **Suffix** parameter, then enter a suffix value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.4.1.30.1

Options

Acceptable values are up to 20 ASCII characters. Embedded null (<NUL >) characters are not allowed. *Default is no characters (disabled)*.

街 Settings	📢 12:37 🛛 🚯
Data Collection	
B Symbology ID, <disable> — Prefix, <> — Suffix, <></disable>	
Change 'Suffix' Setting	
Defaults Refresh	Apply
Symbologies Symbology Optic	ons Beeper
	E

Beeper/LED



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Data Collection icon to access its control panel applet.

Use the right and left arrows to scroll to the **Beeper/LED** tab, then tap this tab to access its parameters.

Most of these functions are not available when using an imager. The following table shows which functions are supported either by an imager or by a laser scanner.

Beeper Function	Imager	Laser Scanner
Beeper Volume	Х	Х
Beeper Frequency		Х
Good Read Beeps		Х
Good Read Beep Duration		Х

The following are parameters for features on the 700 Series Computer. *Note that these are listed in the order of their appearance.*

Beeper Volume

Sets the volume for the good read beep.

Action

Tap the **Beeper volume** parameter, then select an option to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.1.4.1.6.1

Laser Scanner Options

- 0 Low
- 1 High (default)
- 2 Medium
- 3 Off
- 4 Vibrate

B Settings	🕂 12:31 🛛 🚯
Data Collection	
Beeper volume, <high> Beeper frequency, <209 Good read beeps, <1> Good read beep duration Record button, <scanne< td=""><td>0> 1, <80> r></td></scanne<></high>	0> 1, <80> r>
Change 'Beeper volume' Set	ting
Defaults Refresh	Apply
Beeper/LED/Buttons Virtual	Wedge

Imager Options

- Beeper *(default)* Vibrate 1
- 4

🔠 Settings 🛛 📢	9:35	00
Data Collection		
Beeper volume, <vibrate></vibrate>		
Change 'Beeper volume' Setting		-
🖲 Vibrate		
Beeper/LED Imager Virtual Wedge	- F	()
		3 ^

Silencing the Beeper Volume C.

Sounds & Notifications

2 1		
To turn the beeper	r off, tap Start \rightarrow Settings \rightarrow the Personal tab	\rightarrow
Sounds and Notifi	ications \rightarrow the Volume tab, drag the System v	volume
slider bar to the lef	ft "Silent" position, then tap ok to exit this appl	let.

🛃 Setti	ngs	• (€ 9:	50 🚯
Sounds 8	k Notificati	ions	
System Silent	volume	· · · ·	Loud
Enable se Events: Progra No Screer Hardw	ounds for s (warnings, ams otifications (n taps Soft (vare buttons Soft (, system events) alarms, reminder:) Loud ;) Loud	5)
Volume N	otifications		
•		•	•

Beeper Frequency

Sets the frequency for the good read beep. Note that this is not available when you use an imager with your 700 Series Computer.

Action

Tap the **Beeper frequency** parameter, then enter a frequency value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.1.4.1.7.1

Options

1000-4095 (default is 2090)

🛱 Settings	4 € 12:31 🚯		
Data Collection			
Beeper volume, <high> Beeper frequency, <2090> Good read beeps, <1></high>			
- Good read beep duration,	<80>		
ⁱ Record button, <scanner:< td=""><th>> </th></scanner:<>	>		
Change 'Beeper frequency' Se	etting		
2090			
Defaults Refresh Apply			
Beeper/LED/Buttons Virtual V	Vedge 🛛 🕨 🕨		
	▲ 		

Good Read Beeps

Sets the number of good read beeps. Note that this is not available when you use an imager with your 700 Series Computer.

Action

Tap the **Good read beeps** parameter, then select an option to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.1.4.1.8.1

Options

- 0 No beeps
- 1 One beep (default)
- 2 Two beeps

街 Settings	📢 12:33 🕕	
Data Collection		
Beeper volume, <hig <="" <1="" <5ca<="" beecord="" beep="" beeper="" beeps,="" buttop,="" dura="" frequency,="" good="" read="" td=""><th>h> 2090> 2 tion, <80> oper></th></hig>	h> 2090> 2 tion, <80> oper>	
Change 'Good read beeps' Setting		
Defaults Refresh Apply		
Beeper/LED/Buttons Virtual Wedge		
	₩	

Good Read Beep Duration

Sets the duration of the good read beep. Note that this is not available when you use an imager with your 700 Series Computer.

Action

Tap the **Good read beep duration** parameter, then enter a duration value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.1.4.1.9.1

Options

0`2550 Beep duration in milliseconds. (default is 80)

🔠 Settings	📢 12:33 🐽	
Data Collection		
Beeper volume, <high> Beeper frequency, <209 Good read beeps, <1> Good read beeps, <1> Good read beep duration Good read beep duration Cocord button, <scanned< td=""><th>0> , <80> r></th></scanned<></high>	0> , <80> r>	
Change 'Good read beep dur 80	ration' Setting	
Defaults Refresh Apply		
Beeper/LED/Buttons Virtual	Wedge	

Imager



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Data Collection icon to access its control panel applet.

Use the right and left arrows to scroll to the **Imager** tab, then tap this tab to access its parameters.

The following are parameters for the imager. Note that these are listed in the order of their appearance within the Imager tab.

Aimer LED duration

The Aimer LED duration controls the time the Aimer LED is turned on when the scan button is pressed. After this time, images are captured for decoding. The purpose is to position the Aimer LED on the bar code symbol before attempting to decode the bar code. *Note that this is not available when you use a laser scanner with your 700 Series Computer.*

Action

Tap the Aimer LED duration parameter, then enter a value to change this setting. Note that values must be in 50 ms increments, such as 500, 650, or 32500. Values not entered in 50 ms increments will be rounded down. For example, 2489 ms would be rounded down to 2450 ms, 149 ms would be rounded down to 100 ms, etc..

SNMP OID

1.3.6.1.4.1.1963.15.3.3.3.1.1.21.1

Options

0-65500 ms (Default is 0)

B Settings			4 € 10:05	œ
Data Collect	ion			
Aimer LED ⊕- Image dim	duratio ension,	n, <0> <000639	000479>	
Change 'Aime	r LED d	uration' Se	etting ——	
Defaults	Re	fresh	App	ly
Beeper/LED	lmager	Virtual W	/edge	• •
•				₩

Image Dimension

The image dimensions control the horizontal size of the image for decoding. This can restrict the image to one bar code when otherwise, there might be more than one bar code in the image to be decoded. *Note that this is not available when you use a laser scanner with your 700 Series Computer.*

Action

Tap the **Image dimension** parameter, select the position to be changed, then tap an option or enter a value to change this position.

SNMP OID

1.3.6.1.4.1.1963.15.3.3.3.1.1.22.1

Options

Left position	0	Not supported
Right position	0	Not supported
Top position	0-478	Position in pixels (0)
Bottom position	0-479	Position in pixels (479)

B Settings	📢 10:11 🛛 🕕	
Data Collection		
Top position, <not Top position, <0> Bottom position, <4</not 	supported> 79>	
Defaulte Defreeb		
Beeper/LED Imager Virtua		
Virtual Wedge



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Data Collection icon to access its control panel applet.

Use the right and left arrows to scroll to the Virtual Wedge tab, then tap this tab to access its parameters.

The following are parameters for the virtual wedge scanner. *Note that these are listed in the order of their appearance within the Virtual Wedge tab.*

Virtual Wedge

Enables or disables the virtual wedge for the internal scanner. The virtual wedge retrieves scanned Automatic Data Collection (ADC) data and sends it to the keypad driver so that the 700 Series Computer can receive and interpret the data as keypad input.

Because the virtual wedge translates incoming data into keypad input, the size of the keypad buffer limits the effective length of the label to 128 characters. Longer labels may be truncated. For labels of more than 128 characters, you need to develop an application that bypasses the keypad buffer.

Action

Tap the **Virtual Wedge** parameter, then tap an option to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.2.1.1.2.1

Options

- 0 Disable
- 1 Enable (*default*)



Preamble

Sets the preamble that precedes any data you scan with the 700 Series Computer. Common preambles include a data location number or an operator number.

Action

Tap the **Preamble** parameter, then enter a preamble value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.2.1.1.3.1

Syntax

ADdata

where *data* is acceptable values up to 31 ASCII characters. Embedded null (<NUL >) characters are not allowed. *Default is no characters (disabled)*.

🖅 Settings	4 € 1:51	•
Data Collection		
Virtual wedge, <enable> Preamble, <> Postamble, <\t></enable>		
Grid, <>		
iCode page, <1252>		
Change 'Preamble' Setting ——		
Defaults Refresh	Apply	<u> </u>
Beeper/LED Imager Virtual W	edge	• •
		\$1▲



Note: When you enter the AD command without data, the preamble is disabled. If you want to use quotation marks or the following combinations of characters as part of the appended data, separate those characters from the AD command with quotes. If you do not use quotes as described here, the 700 Series Computer will interpret the characters as another configuration command:

- AD AE
- AF
- KC
- BV
- EX
- DF

EXAMPLE:

To use the two-character string BV as a preamble, scan this command (as a Code 39 label) or send this command through the network: \$+AD"BV"

Postamble

Sets the postamble that is appended to any data you scan with the 700 Series Computer. Common postambles include cursor controls, such as tabs or carriage return line feeds.

Action

Tap the **Postamble** parameter, then enter a postamble value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.2.1.1.4.1

Syntax

AEdata

where *data* is any acceptable values up to 31 ASCII characters. Embedded null (<NUL >) characters are not allowed. *Default is the tab character* (\t).

街 Settings	🕂 12:28 🐽	
Data Collection		
i Virtual wedge, <enable></enable>		
Preamble, <>		
Postamble, <\t>		
Grid, <>		
Code page, <1252>		
Change 'Postamble' Setting -		
\t		
Defaults <u>R</u> efresh	Apply	
Beeper/LED Imager Virtual	Wedge	
	E E I	



Note: When you enter the AE command without data, the postamble is disabled. If you want to use quotation marks or the following combinations of characters as part of the appended data, separate those characters from the AE command with quotes. If you do not use quotes as described here, the 700 Series Computer will interpret the characters as another configuration command.

- AD AE
- AF
- KC
- BV EX
- DF

EXAMPLE:

To use the two-character string BV as a postamble, scan this command (as a Code 39 label) or send this command through the network: \$+AE`BV"

Grid

Sets the virtual wedge grid, which filters the data coming from this 700 Series Computer. The data server supports data filtering, which allows you to selectively send scanned data. The virtual wedge grid is similar to the "format" argument of the C Runtime Library scan function.

Action

Tap the Grid parameter, then enter a grid value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.2.1.1.5.1

Syntax

AF<symID> filter-expression= > editing-expression where:

- *<symID>* The AIM symbology ID.
- filter-expression

Any character string that includes valid filter expression values, and editing-expression is any character string that includes valid editing expression values.

• *<width>*

Any positive integer or NULL. A NULL width means that the field type (defined next) applies all the way to the end of the data string. A non-NULL width means that the field applies to that many characters of data. The grid can be up to 240 characters in length. *Default is NULL*.

😰 Settings	4 € 1:51 🛛 🔂
Data Collection	
 Virtual wedge, <enable></enable> Preamble, <> Postamble, <\t> Grid, <> Code page, <1252> 	
Change 'Grid' Setting	
Defaults Refresh	Apply
Beeper/LED Imager Virtual W	/edge
	∎ ►

Code Page

Sets the virtual wedge code page. The code page controls the translation from the character set of the raw collected data to Unicode, which is the character set expected by Windows CE applications. The default code page is 1252, which is the Windows Latin 1 (ANSI) character set.

Action

Tap the **Code Page** parameter, then select an option to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.15.3.2.1.1.6.1

Options

The only acceptable value for the code page parameter is "1252," which is the default.

街 Settings	4 € 1:51 🚯
Data Collection	
Virtual wedge, <enable></enable>	
Preamble, <>	
Postamble, <\t>	
Grid, <>	
Code page, <1252>	
Change 'Code page' Setting -	
1252	
1202	
<u>Deraults</u> <u>Refresh</u>	Apply
Beeper/LED Imager Virtual	Wedge
	₩

SNMP Control Panel Applet

Simple Network Management Protocol (SNMP) parameters include identification information, security encryption, security community strings, and traps.



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings

 \rightarrow the System tab \rightarrow the SNMP icon to access its control panel applet.

83 Settings	4 € 6:43 0 8
SNMP	
Read Only Community, <p< td=""><td>oublic></td></p<>	oublic>
Read/Write Community, <	private>
Write Encryption, <off></off>	
Encryption Key, <>	
Change 'Read Only Communit	y' Setting
public	
Defaults Refresh	Apply
Security Traps Identification	۱
	■ -

Tap a tab to access its menus. These tabs represent three groups of settings or parameters:

- Security (starting on the next page)
- Traps (starting on page 336)
- Identification (starting on page 338)

Security



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the SNMP icon \rightarrow the Security tab to access its parameters.

The following are parameters that affect encryption and community strings. *Note that these are listed in the order of their appearance within the Security tab.*

Read Only Community

Sets the read-only community string for this 700 Series Computer, which is required for processing of SNMP get and get next requests.

Action

Tap the **Read Only Community** parameter, then enter a community string to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.10.5.1.2.0

Options

The read-only community string can be up to 128 ASCII characters. *De-fault is Public*.

🔠 Settings	€ 6:43 🐽	
SNMP		
Read Only Community, <pu Read/Write Community, <pr< td=""><td>olic> ivate></td></pr<></pu 	olic> ivate>	
Encryption Key, <>		
Change 'Read Only Community' Setting		
public		
Defaults Refresh	<u>A</u> pply	
Security Traps Identification		

Read/Write Community

Sets the read/write community string, which is required for processing of SNMP set requests by this 700 Series Computer. An SNMP packet with this name as the community string will also process SNMP get and next requests.

Action

Tap the **Read/Write Community** parameter, then enter a community string to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.10.5.1.3.0

Options

The read/write community string can be up to 128 ASCII characters. *Default is Private.*

🔠 Settings	📢 6:43 🛛 🔂
SNMP	
Read Only Community, <pu< td=""><td>ublic></td></pu<>	ublic>
Read/Write Community, <p< td=""><td>private>.</td></p<>	private>.
Read Encryption, <off></off>	
Write Encryption, <off></off>	
i Encryption Key, <>	
Change 'Read/Write Communit	y' Setting —
private	
Defaults Refresh	Apply
Security Traps Identification	
	E

Read Encryption

Sets the packet-level mode of security for SNMP read-only requests. If you enable read encryption, all received SNMP get and get next packets have to be encrypted or the packet will not be authorized. If encryption is enabled, you can only use software provided by Intermec Technologies.



Note: To enable security encryption, you also need to set the Security Encryption Key (page 335).

Action

Tap the **Read Encryption** parameter, then select an option to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.10.5.1.4.0

Options

- 1 On SNMP get and get next packets must be encrypted
- 2 Off SNMP packets do not have to be encrypted (*default*)

Settings		📢 6:43 🐽	
SNMP			
Read Only	Community, <pu< td=""><td>iblic></td></pu<>	iblic>	
Read/Writ	e Community, <p< td=""><td>rivate></td></p<>	rivate>	
- Read Encr	yption, <off></off>		
Write Encr	yption, <0ff>		
Encryption	n Key, <>		
Change 'Read	Change 'Read Encryption' Setting		
¯ () On — — — — — — — — — — — — — — — — — —			
0	ff		
Defaults	Refresh	Apply	
Security Trap	s Identification		
		■	

Write Encryption

Sets the packet-level mode of security for SNMP read/write requests. If you enable write encryption, all SNMP packets that are received with the read/write community string have to be encrypted or the packet will not be authorized. You need to use software from Intermec Technologies that supports encryption.



Note: To enable security encryption, you also need to set the Security Encryption Key (page 335).

Action

Tap the Write Encryption parameter, then select an option to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.10.5.1.5.0

Options

- 1 On SNMP packets must be encrypted
- 2 Off SNMP packets do not have to be encrypted (*default*)

🔠 Settings		€ 6:43 🚯	
SNMP			
Read Only Con	nmunity, <pu immunity, <pi< td=""><td>blic> rivate></td></pi<></pu 	blic> rivate>	
Read Encryptic	on, <off></off>		
Encryption Key	Encryption, <op>Encryption Key, <></op>		
Change 'Write Encryption' Setting			
🔘 🔘 Off			
Defaults	<u>R</u> efresh	Apply	
Security Traps I	dentification		
		₩	

Encryption Key

Identifies the key that this 700 Series Computer uses to encrypt or decipher SNMP packets. Encryption is used only by software provided by Intermec Technologies. If encryption is enabled, SNMP management platforms will not be able to communicate with the 700 Series Computer. The encryption key is returned encrypted.

Action

Tap the Encryption Key parameter, then enter a security encryption key value to change this parameter setting.



Note: You also need to set either Read Encryption (page 333) or Write Encryption (page 334) or both.

SNMP OID

1.3.6.1.4.1.1963.10.5.1.6.0

Options

The encryption key can be from 4 to 20 ASCII characters. *Default is NULL*.

街 Settings		€ 6:43 🐽
SNMP		
Read Only Co Read/Write Co Read Encrypti Write Encrypti Write Encrypti	mmunity, <pul ommunity, <pr on, <off> ion, <off> v, <></off></off></pr </pul 	olic> ivate>
Change Encryptic	on Key' Setting	
Defaults	<u>R</u> efresh	Apply
Security Traps	Identification	

Traps



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the SNMP icon \rightarrow the Traps tab to access its parameters.

The following are authentication and threshold parameters for traps. *Note that these are listed in the order of their appearance within the Traps tab.*

Authentication

Determines whether to send authentication traps. When trap authentication is enabled, an authentication trap is sent if an SNMP packet is received by the master agent with an invalid community string.

Action

Tap the Authentication parameter, then select an option to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.10.5.2.2.0

Options

1 On (default)

2 Off

🔠 Settings	4 € 6:43 🔞
SNMP	
Authentication, <on> Threshold, <10></on>	
Change 'Authentication' Setting]]
Off	
Defaults Refresh	Apply
Security Traps Identification	

Threshold

Determines the maximum number of traps per second that the master agent generates. If the threshold is reached, the trap will not be sent.

Action

Tap the **Threshold** parameter, then enter a threshold value to change this parameter setting.

SNMP OID

1.3.6.1.4.1.1963.10.5.2.3.0

Options

Any positive integer value. Default is 10.

🕖 Settings	4 € 6:44 😡
SNMP	
Threshold, <10>	
Change 'Threshold' Setting	
Defaults Refresh	Apply
Security Traps Identification	
	=

Identification



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the SNMP icon \rightarrow the Identification tab to access its parameters.

The following are parameters for contact, location, and name information for support purposes. *Note that these are listed in the order of their appearance within the Identification tab.*

Contact

Sets the contact information for the person responsible for this 700 Series Computer.

Action

Tap the **Contact** parameter, then enter the name of your contact representative to change this parameter setting.

SNMP OID

1.3.6.1.2.1.1.4.0

Options

The identification contact may be up to 255 ASCII characters. *Default is no characters or blank.*

🛃 Settings	🕂 6:44 🛛 😡
SNMP	
Contact, <add administr<br="" an="">Name, <pocket_pc> Location, <add a="" if<="" location="" td=""><th>ator here></th></add></pocket_pc></add>	ator here>
Change 'Contact' Setting	here
Defaults Refresh	Apply
Security Traps Identification	

Name

Sets the assigned name for this 700 Series Computer.

Action

Tap the Name parameter, then enter the name of your 700 Series Computer to change this parameter setting.

SNMP OID

1.3.6.1.2.1.1.5.0

Options

The identification name may be up to 255 ASCII characters. *Default is no characters or blank*.

🔠 Settings	📢 6:44 🛛 🚯
SNMP	
Contact, <add admin<br="" an="">Name, <pocket_pc> Location, <add a="" locatio<="" td=""><td>istrator here> n if applicable></td></add></pocket_pc></add>	istrator here> n if applicable>
Change 'Name' Setting Pocket_PC	
Defaults Refresh	Apply
Security Traps Identificati	

Location

Sets the identification location for this 700 Series Computer, such as "Shipping."

Action

Tap the **Location** parameter, then enter the location of where your 700 Series Computer to change this parameter setting.

SNMP OID

1.3.6.1.2.1.1.6.0

Options

The identification location may be up to 255 ASCII characters. *Default is no characters or blank.*

😰 Settings 🛛 📢 🚱
SNMP
Contact, <add administrator="" an="" here=""></add>
Location, <add a="" applicable="" if="" location=""></add>
Change 'Location' Setting
Add a location if applicable
Defaults Refresh Apply
Security Traps Identification
E

Unit Information Control Panel Applet

Unit Information is a read-only control panel applet that provides information about your 700 Series Computer, such as software version builds, available CAB files, and the internal battery status.

This control panel applet is only available in the 700 Series Computer if Intermec Content is enabled, the Plus region is enabled and installed, and a laser scanner is installed.



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Unit Information icon to access its control panel applet.

Bettings	📢 9:30 🛛 🔂
Unit Information	
700 Platform Build, <1.	29>
DataCollection Build, <s< td=""><td>earayimg003.045></td></s<>	earayimg003.045>
Defaults Refresh	Apply
Versions Battery Status	CAB Files
	₩

Tap a tab to access its menus. These tabs represent three groups of settings or parameters:

- Versions (starting on the next page)
- Battery Status (starting on page 343)
- CAB Files (starting on page 344)

Versions



You can view the latest software build version on your 700 Series Computer by accessing the **Unit Information** control panel applet.

To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Unit Information icon \rightarrow the Versions tab to view the latest software build version. Tap ok to exit this information.

8 Sett	ings	ŧ	9:30	•
Unit Inf	ormation			
700 F S9C, Data	Ylatform Build,<1 <> Collection Build,·	29> <searayimg< td=""><td>003.049</td><td>ō></td></searayimg<>	003.049	ō>
Defaults	<u>R</u> efres	<u>h</u>	<u>A</u> pply	
Versions	Battery Status	CAB Files		-1
			***	8 ^

Below are some of the software applications you may find on this screen:

• 700 Platform Build:

Shows the latest development or released version of the software build for the 700 Series Computer.

• S9C:

Provides the name and version of the scanner file built into this 700 Series Computer, along with the current CPU version.

• DataCollection Build:

Shows the latest development or released version of the software build for the Data Collection control panel applet.

Battery Status

You can view the battery status for your 700 Series Computer by accessing the Unit Information control panel applet. Unit Manager applications are available on the *700 Series Color Unit Manager CD-ROM*. For more information, consult your Intermec sales representative.



To access the settings from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Unit Information icon \rightarrow the Battery Status tab to view the current status. Tap ok to exit this information.

🔠 Settings	€	0:44	ok
Unit Information			
Main battery, <good -="" 63%<="" th=""><th>></th><th></th><th></th></good>	>		
<u>D</u> efaults <u>R</u> efresh		Appl	٧
Versions Battery Status CAB	Files		
			\$

CAB Files

You can view the latest developer or released version of each CAB file from Intermec Technologies Corporation that are installed in your 700 Series Computer via the **Unit Information** control panel applet. *Custom CAB files are not displayed in this applet.* See the *Software Tools User's Manual* for more information about these files.



To access the information from the 700 Series Computer, tap Start \rightarrow Settings \rightarrow the System tab \rightarrow the Unit Information icon \rightarrow the CAB Files tab to view the current CAB file versions. Tap ok to exit this information.

🛃 Settings	4 € 11:00 🚳
Unit Information	
BtMainStack, <1.2.7.32> 	g001.021> ayimg001.021>
Defaults Refresh	Apply
Versions Battery Status CA	.B Files

When a CAB file is built, a registry entry is created with a build number for that file. This CAB Files control panel applet looks for a registry key for each CAB file installed. When the registry entry is found, the CAB file name and version number information are displayed. If a CAB file has not been installed, then its information is not displayed.

Below is a list of CAB files from Intermec Technologies that are available for your 700 Series Computer with their latest developer or released version of the software build. Should you need to add any of these to your 700 Series Computer, contact an Intermec representative.

• BtMainStack:

Installation of the Main Bluetooth Stack is handled automatically as part of the operating system boot-up procedure. *See Chapter 4, "Network Support," for more information about Bluetooth wireless printing.*

• Comm Port Wedge:

The software build for the Comm Port Wedge. Note that the Comm Port Wedge CAB file is available on the 700C Tools CD.

• NPCPTest:

This installs a Norand[®] Portable Communications Protocol (NPCP) Printing test application which will print to an Intermec[®] 4815, 4820, or 6820 Printer. *See Chapter 5, "Printer Support," for more information.*



• PDWPM0C:

This is the installer for the Wireless Printing Demo application. To run this demonstration, tap Start \rightarrow Programs \rightarrow the Wireless Printing Demo icon. *Press Help in the demo application for more information*.

• S9C Upgrade:

Installs the files needed to upgrade the S9C scanner firmware. See the Recovery CD Help for more information about upgrading the firmware.

• SDK:

Installs the Intermec Software Developer's Kit (SDK). See the SDK User's Manual for more information.

• Unit Manager:

Installs the Unit Manager application which provides tools for remotely managing the 700 Series Computer. Unit Manager applications are available on the *700 Series Color Unit Manager CD-ROM*. For more information, consult your Intermec sales representative.

• Unit Manager Help:

Installs the online help for the Unit Manager application.

• WinCfg:

Configures the NRINET.INI file, launches the NRINet client, and loads and unloads the LAN and WLAN device drivers. *See the Windows* 95 and Windows CE Configuration Utilities Reference Manual (P/N: 978-054-010) for more information.

• Wireless Printing Sample:

Installs a sample application that developers can use for reference when they are developing their own Wireless Printing applications. The source code for this application is included as part of the Wireless Printing SDK on the 700C Tools CD. *See the SDK User's Manual for more information.*

• ActiveX Control Tools:

This lists some of the CAB files that may be available with which to install ActiveX Control Tools. *See the SDK Online Help for more information*.

• AXCommunication:

Communication controls that transmit or receive messages from input connections.

• AXFileTransfer:

File transfer controls that transmit and receive files using the Trivial File Transfer Protocol (TFTP).

• AXReaderCommand:

Reader command functions that modify and retrieve configuration information from your 700 Series Computer.

• AXVWedge:

The virtual wedge control that retrieves scanned ADC data and sends it to the keyboard driver to interpret data as keyboard input.

Appendix A — Control Panel Applets



Configuration parameters are also configurable using a Unit Manager application which accesses the 700 Series Computer through a web browser on your desktop PC via the SRDEVMGMT.CAB file.

Unit Manager applications are available on the 700 Series Color Unit Manager CD-ROM. For more information, consult your Intermec sales representative.



Note: Parameter information, such as SNMP OID and options, is detailed in Appendix A, "*Control Panel Applets*."

Data Collection



Within the Unit Manager, click **Configuration** from the left navigation bar, then click the **Data Collection** icon to access any of these tabs: Symbologies, Symbology ID, Beeper/LED, or Virtual Wedge.

Symbologies

Within the Unit Manager, select Configuration Management \rightarrow Data Collection, then click the Symbologies tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Codabar (page 292)
- Code 11 (page 306)
- Code 128 (page 295)
 - Code 128 Options (page 296)
 - Code 128 FNC1 Character (page 297)
- Code 39 (page 290)
- Code 93 (page 294)
 - Code 93 Length (page 294)
- Data Matrix (page 308)
- Interleaved 2 of 5 (page 303)
- Matrix 2 of 5 (page 304)
- MSI (page 299)
- PDF 417 (page 300)
 - Macro PDF (page 300)
 - Micro PDF 417 (page 302)
- Plessey (page 298)
- QR Code (*page 307*)
- Standard 2 of 5 (page 291)
- Telepen (page 305)
- UPC/EAN (page 293)

Symbology ID

Within the Unit Manager, select Configuration Management \rightarrow Data Collection, then click the Symbology ID tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Prefix (page 315)
- Suffix (page 316)
- Symbology ID (page 309)
 - Codabar user ID (page 310)
 - Code 11 user ID (page 314)
 - Code 128 user ID (page 310)
 - Code 39 user ID (*page 310*)
 - Code 93 user ID (page 311)
 - EAN-13 user ID (*page 313*)
 - EAN-8 user ID (page 313)
 - Interleaved 2 of 5 user ID (page 311)
 - Matrix 2 of 5 user ID (page 313)
 - MSI user ID (page 311)
 - PDF 417 user ID (page 311)
 - Plessey user ID (page 312)
 - Standard 2 of 5 user ID (page 312)
 - Telepen user ID (page 313)
 - UPC-A user ID (page 312)
 - UPC-E user ID (page 312)

Beeper/LED

Within the Unit Manager, select Configuration Management \rightarrow Data Collection, then click the Beeper/LED tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Beeper Frequency (page 320)
- Beeper Volume (page 318)
- Good Read Beep Duration (page 322)
- Good Read Beeps (page 321)

Imager

Within the Unit Manager, select **Configuration Management** \rightarrow **Data Collection**, then click the **Imager** tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Aimer LED Duration (page 323)
- Image Dimension (page 324)

Virtual Wedge

Within the Unit Manager, select Configuration Management \rightarrow Data Collection, then click the Virtual Wedge tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Code Page (page 329)
- Grid (page 328)
- Postamble (page 327)
- Preamble (page 326)
- Virtual Wedge (page 325)

SNMP



SNMP

Within the Unit Manager, click **Configuration** from the left navigation bar, then click the **SNMP** icon to access any of these tabs: Security, Traps, or Identification.

Security

Within the Unit Manager, select **Configuration Management** \rightarrow **SNMP**, then click the **Security** tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Encryption Key (page 335)
- Read Encryption (page 333)
- Read Only Community (page 331)
- Read/Write Community (page 332)
- Write Encryption (page 334)

Traps

Within the Unit Manager, select **Configuration Management** \rightarrow **SNMP**, then click the **Traps** tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Authentication (page 336)
- Threshold (page 337)

Identification

Within the Unit Manager, select **Configuration Management** \rightarrow **SNMP**, then click the **Identification** tab to access the following parameters. Options for these parameters are listed on the page provided. *These are listed in alphabetical order*.

- Contact (page 338)
- Location (page 340)
- Name (*page 339*)

Unit



Within the Unit Manager, click **Configuration** from the left navigation bar, then click the **Unit** icon to access any of these tabs: Date/Time, Display, Keypad, Power Management, or Speaker.

Date/Time

Sets the current date and time.

Action

Click the **Date/Time** tab, then select **Date** or **Time** and make changes in the entry field, or tap (+) to expand either the Date or Time parameter, select the setting to be changed, then select a value from the drop-down list or enter a new value to change this setting.

SNMP OID

Date:	1.3.6.1.4.1.1963.15.501.2.1.0
Time:	1.3.6.1.4.1.1963.15.501.2.2.0

Options

Year	0000`999	(1999)
Month	1-12 (6)	
Day	1-31 (1)	
Hour	0-23 (0)	
Minute	0-59 (00)	
Second	0-59 <i>(00)</i>	
	Year Month Day Hour Minute Second	Year0000`999Month1-12 (6)Day1-31 (1)Hour0-23 (0)Minute0-59 (00)Second0-59 (00)

Linit		5
Unit		
Configure	Date/Time Display Keypad Power Management Speaker	
	10.10.111.9	Default
Data Collection	EPDate <20020617> ▲ 1 EPTime <112605>	Refresh
Unit		
\square		
SNMP		
		Annly

Backlight Timeout

Sets the length of time that the display backlight remains on. If you set a longer timeout value, you use the battery power at a faster rate.

Action

Click the **Display** tab, then select an option from the **Backlight timeout** drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.13.1.0

Syntax

```
DFdata
```

where *data* is any of the following:

- 10 10 seconds
- 30 30 seconds
- 60 1 minute (*default*)
- 120 2 minutes
- 180 3 minutes
- 240 4 minutes
- 300 5 minutes

Intermec	Configuration Management	
Unit		\square
Configure	Date/Time Display Keypad Power Management Speaker	
Data Collection	10.10.111.9	Default
		Reiresn
Unit		
SNMP		
-		
-		
Troubleshoot		Apply

Key Clicks

Enables or disables the keypad clicks. The 700 Series Computer emits a click each time you press a key or decode a row of a two-dimensional symbology.

Action

Click the Keypad tab, then select an option from the Key clicks dropdown list.

SNMP OID

1.3.6.1.4.1.1963.15.12.1.0

Syntax

KCdata where *data* is any of the following:

- 0 Disable clicks
- 1 Enable soft key clicks
- 2 Enable loud key clicks (default)

Intermed	Configuration Management
Unit	
Configure	Date/Time Display Keypad Power Management Speaker 10.10.111.9 Default Key clicks <loud> Refresh</loud>
Troubleshoot	Apply

Automatic Shutoff

Sets the length of time the 700 Series Computer remains on when there is no activity. When you turn on the 700 Computer, it either resumes exactly where it was when you turned it off or boots and restarts your application.

Action

Click the **Power Management** tab, then select an option from the **Auto-matic shutoff** drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.11.3.0

Syntax

EZdata where *data* is any of the following:

- 1 1 minute
- 2 2 minutes
- 3 3 minutes (default)
- 4 4 minutes
- 5 5 minutes

Intermec Configuration Management			
Unit			
Configure	Date/Time Display Keypad Power Management Speaker		
	10.10.111.9 Default		
	Refresh		
Unit			
014101			
-			
			
Troubleshoot	Apply		

Volume

Changes the volume of all audio signals.

Action

Click the **Speaker** tab, then select an option from the **Volume** drop-down list.

SNMP OID

1.3.6.1.4.1.1963.15.3.1.3.0

Syntax

```
BVdata
```

where *data* is any of the following:

- 0 Off
- 1 Very quiet
- 2 Quiet
- 3 Normal (default)
- 4 Loud
- 5 Very loud

Intermec Configuration Management			
Unit			
Configure	Date/Time Display Keypad Power Management Speaker		
Data Collection	10.10.111.9 Default ──Volume <off> ▲ Refresh</off>		
Unit			
SNIMP			
	•		
Troubleshoot	Apply		

Using Reader Commands

After the 700 Series Computer is connected to your network, you can send the 700 Series Computer a reader command from an application to perform a task, such as changing the time and date. Some reader commands temporarily override the configuration settings and some change the configuration settings.

Change Configuration

The Change Configuration command must precede any configuration command. If you enter a valid string, the 700 Series Computer configuration is modified and the computer emits a high beep. To send the Change Configuration command through the network, use the \$+ [command] syntax where command is the two-letter command syntax for the configuration command followed by the value to be set for that command.

You can also make changes to several different commands by using the + [command]...[command n] syntax. There are seven configuration command settings that can be changed in this way. See each command for information on respective acceptable "data" values.

Command	Syntax
Audio Volume	BV <i>data</i>
Automatic Shutoff	EZdata
Backlight Timeout	DF <i>data</i>
Key Clicks	KCdata
Virtual Wedge Grid	AFdata
Virtual Wedge Postamble	AEdata
Virtual Wedge Preamble	ADdata



Note: See Appendix A, "*Control Panel Applets*" for more information about the Virtual Wedge Postamble and Virtual Wedge Preamble commands.

Example 1

To change the Beep Volume to Off, you can send this string to the 700 Series Computer through the network: \$+BV0

where:

- \$+ Indicates Change Configuration.
- BV Specifies the Audio Volume parameter.
- 0 Specifies a value of Off.

Example 2

To change the Beep Volume to Very Quiet and the Virtual Wedge Grid to 123: \$+BV1AF123

where:

\$+	Indicates Change Configuration
BV1	Specifies Audio Volume, set to Very Quiet (1)
AF123	Specifies Virtual Wedge Grid, set to a value of 123

Set Time and Date

This command sets the date and time on the 700 Series Computer. The default date and time is *June 1, 1999 at 12:00 AM*.

From the network, send the following:

/+ yyyymmddhhmmss

where acceptable values for the date are:

УУУУ	0000-9999 Year	
mm	01-12	Month of the year
dd	01-31	Day of the month
hh	00-23	Hour
mm	00-59	Minutes
SS	00-59	Seconds



You can also set the time and date by using Configuration Management in Unit Manager, or by using the **Clock** control panel applet in the Settings menu. To access this control panel applet, tap **Start** \rightarrow **Settings** \rightarrow the **System** tab \rightarrow the **Clock** icon to access its control panel applet.

🛃 Settings	↓ € 9:4	7 🚯	
Clock			
Home			
. 12 .	GMT-8 Pacific US	•	
 g~~~~~ 3	9 :47:53 AM	▲ ▼	
· ·	7 / 2 /2002	•	
°6 °			
◯ Visiting			
. 12 .	GMT+1 Paris,Madr	id 🔻	
9 + 3	6 :47:53 PM	▲ ▼	
[[™] .]	7 / 2 /2002	•	
°6°			
Time Alarms			



This appendix contains a brief explanation of some of the bar code symbologies that the 700 Series Color (700C) Mobile Computer decodes and explains some of the general characteristics and uses of these bar code types. It also includes several bar code labels that can be scanned into your 700 Series Computer.

Bar Code Symbologies

Specific bar code algorithms can be enabled using the setup menus or the host computer. Once the computer correctly decodes a bar code, the computer encodes data with descriptive information about the symbol. Response time is improved by limiting the computer to the bar codes being used.

Data Bar Code Type	Data Format	Data Length
UPC short (UPC-E)	ndddddc	8
EAN short (EAN-8)	fnddddc	8
UPC long (UPC-A)	nddddddddc	12
EAN long (EAN-13)	fnddddddddc	13
UPC short add-on 2	ndddddcaa	10
EAN short add-on 2	fnddddcaa	10
UPC long add-on 2	nddddddddcaa	14
EAN long add-on 2	fnddddddddcaa	15
UPC short add-on 5	ndddddcaaaaa	13
EAN short add-on 5	fnddddcaaaaa	13
UPC long add-on 5	nddddddddcaaaaa	17
EAN long add-on 5	fnddddddddcaaaaa	18
Interleaved 2 of 5	dd	Scan device dependent
Standard 2 of 5	dd	Scan device dependent
Plessey	ddc	Scan device dependent
Codabar	sdds	Scan device dependent
Code 11	dd	Scan device dependent
Code 39	dd	Scan device dependent
Extended Code 39	dd	Scan device dependent
Code 93	dd	Scan device dependent
Code 128	dd	Scan device dependent

Bar Code Data String Formats



Note: These bar code data definitions apply to the Data Format column in the previous table

- a Add-on code digits
- c Check digits
- d Bar code digits
- f EAN flag 1 characters
- n Number system digits
- s Start and stop digits

If MOD 10 or MOD 11 check digits are enabled, the digit falls at the end of a bar code data string. Each check digit enabled extends the bar code data string length by one character.
The 700 Series Computer recognizes eleven of the most widely used bar code symbologies. With bar code symbologies, like languages, there are many different types. A bar code symbology provides the required flexibility for a particular inventory tracking system.

A symbology may be for particular industries, such as food and beverage, automotive, railroad, or aircraft. Some of these industries have established their own bar code symbology because other symbologies did not meet their needs.

Without going into great detail on the bar code structure, note that no two products use the same bar code. Each product gets a unique bar code.

Industries that use a particular type of bar code symbology have formed regulating committees or are members of national institutes that issue and keep track of bar codes. This ensures that each organization that contributes to a particular industry conforms to its standard. Without some form of governing body, bar coding would not work.

- UPC (Universal Product Code) with/without add-ons
- EAN (European Article Numbering Code) with/without add-ons
- Codabar
- C11 (Code 11)
- C39 (Code 39)
- C93 (Code 93)
- C128 (Code 128)
- I 2 of 5 (Interleaved 2 of 5 Code)
- S 2 of 5 (Standard 2 of 5)
- Plessey
- MSI (a variant of Plessey)

UPC	
	The UPC (Universal Product Code) is the symbology used throughout the grocery and retail industries. This bar code symbology contains two pieces of numerical information encoded on the bar code, producer identification, and product identification information.
	The UPC symbol is 12 characters long. The first character of the UPC symbol is a number system character, such as "0" for grocery items and "3" for drug- and health-related items.
	The UPC symbology is for retail environments such as grocery stores, convenience stores, and general merchandise stores.
	Some retail items are so small that a standard UPC bar code cannot fit on the packaging. When this occurs there is a permitted shorter version of the UPC symbology, referred to as UPC-E. UPC-E is six characters long (eight including number system and check digit), approximately half the size of a standard UPC bar code.
EAN	
	EAN (European Article Numbering) symbology is similar to UPC symbol- ogy, except that it contains 13 characters and uses the first two to identify countries.
	The EAN symbology is used in the retail environment throughout most of Europe. Though similar to UPC symbology, these are not interchangeable.
Codabar	
	Codabar was for retail price-labeling systems. Today it is widely accepted by libraries, medical industries, and photo finishing services.
	Codabar is a discrete, self-checking code with each character represented by a stand-alone group of four bars and three intervening spaces.
	Four different start or stop characters get defined and designated "a", "b", "c", and "d". These start and stop characters are constructed using one wide bar and two wide spaces. A complete Codabar symbol begins with one of the start or stop characters followed by some number of data char- acters and ending in one of the start or stop characters.
	Any of the start or stop characters may be used on either end of the symbol. It is possible to use the 16 unique start or stop combinations to identify label type or other information.
	Since Codabar is variable-length, discrete, and self-checking, it is a versatile symbology. The width of space between characters is not critical and may vary significantly within the same symbol. The character set consists of "0" through "9", "-", "\$", ":", "/", ".", and "+".
	The specific dimensions for bars and spaces in Codabar optimize perfor- mance of certain early printing and reading equipment. Codabar has 18 different dimensions for bar and space widths. So many different dimen- sions often result in labels printed out of specification and cause Codabar printing equipment to be more expensive.

Code 11	
	Code 11 satisfies the requirements for a very high density, discrete numeric bar code. The name Code 11 derives from 11 different data characters that can be represented, in addition to a start or stop character.
	The character set includes the 10 digits and the dash symbol. Each charac- ter is represented by a stand-alone group of three bars and two intervening spaces. Although Code 11 is discrete, it is not self-checking. A single print- ing defect can transpose one character into another valid character. One or two check digits obtain data security.
	The specifications for Code 11 suggest that this code should have a narrow element width of 7.5 mils. This results in an information density of 15 characters per inch.
Code 39	
	Code 39 (C39) is the most widely used symbology among the industrial bar codes. Most major companies, trade associations, and the federal gov- ernment find this code to fit their needs. The main feature of this symbol- ogy is the ability to encode messages using the full alphanumeric character set, seven special characters, and ASCII characters.
	Programming for this symbology can be for any length that the application requires. The application program for the 700 Series Computer handles symbology at least one character but no more than 32 characters in length.
	When programming the computer for Code 39, it is important to set the symbology limit as close as possible (minimum and maximum bar code lengths being scanned). Doing so keeps the computer bar code processing time to a minimum and conserves battery power.
	Bar code readers can respond to Uniform Symbology Specification symbols in non-standard ways for particular applications. These methods are not for general applications, because of the extra programming required. Code 39 Full ASCII is one example of non-standard code.
E	Note : See page 368 to scan several Code 39 bar code labels available to change settings on your 700 Series Computer.
Encoded Code 39 (Co	ncatenation)
	If the first data character of a symbol is a space, the reader may be pro- grammed to append the information contained in the remainder of the symbol to a storage buffer. This operation continues for all successive sym-

symbol to a storage buffer. This operation continues for all successive symbols that contain a leading space, with messages being added to the end of previously stored ones. When a message is read which does not contain a leading space, the contents are appended to the buffer, the entire buffer is transmitted, and the buffer is cleared.

Encoded Code 39 (Full ASCII)

If the bar code reader is programmed for the task, the entire ASCII character set (128 characters) could be coded using two character sequences: a symbol ("\$",".","%","/") followed by a letter of the alphabet.

Code 93

	The introduction of Code 93 provided a higher density alphanumeric symbology designed to supplement Code 39. The set of data characters in Code 93 is identical with that offered with Code 39. Each character con- sists of nine modules arranged into three bars and three spaces.
	Code 93 uses 48 of the 56 possible combinations. One of these characters, represented by a square, is reserved for a start or stop character, four are used for control characters, and the remaining 43 data characters coincide with the Code 39 character set. An additional single module termination bar after the stop character concludes the final space.
	Code 93 is a variable length, continuous code that is not self-checking. Bar and spaces widths may be one, two, three, or four modules wide. Its struc- ture uses edge-to-similar-edge decoding. This makes the bar code immune to uniform ink spread, which allows liberal bar width tolerances.
	Code 93 uses two check characters. Its supporters believes this makes it the highest density alphanumeric bar code. The dual check digit scheme provides for high data integrity. All substitution errors in a single character are detected for any message length.
Code 128	
	Code 128 (C128) is one of the newest symbologies used by the retail and manufacturing industries. It responds to the need for a compact alphanumeric bar code symbol that could encode complex product identification.
	The fundamental requirement called for a symbology capable of being printed by existing data processing printers (primarily dot-matrix printers) that produce daily, work-in-progress, job, and product traceability docu- ments. The ability to print identification messages between 10 and 32 characters long, on existing forms and labels deemed an important require- ment.
	Code 128 uniquely addresses this need as the most compact, complete, alphanumeric symbology available.
	Additionally, the Code 128 design with geometric features, improves scan- ner read performance, does self-checking, and provides data message man- agement function codes.
	Code 128 encodes the complete set of 128 ASCII characters without ad- ding extra symbol elements. Code 128 contains a variable-length symbolo- gy and the ability to link one message to another for composite message transmission. Code 128, being a double-density field, provides two numer- ic values in a single character.
	Code 128 follows the general bar code format of start zone, data, check digit, stop code, and quiet zone. An absolute minimum bar or space dimension of nine mils (0.010 inch minimum nominal \pm 0.001 inch tolerance) must be maintained.
	Characters in Code 128 consist of three bars and three spaces so that the total character set includes three different start characters and a stop character.

UCC/EAN-128 Shipping Container Labeling is a versatile tool that can ease movement of products and information. The Shipping Container Labeling bar code can take any form and usually has meaning only within the company or facility where applied.

Because this *random* data can get mistaken later for an industry standard code format, the UCC and EAN chose a symbology uniquely identified from these other bar codes. This standard is for maximum flexibility, to handle the diversity of distribution in global markets by cost efficiency.

The UCC/EAN-128 Container Labeling specification calls for a FUNC1 to immediately follow the bar code's start character. FUNC1 also follows any variable-length application field. The specification also calls for the computer to send "]C1" for the first FUNC1. The specification requires that the computer send a "<GS>" (hex 1D) for subsequent FUNC1 codes in the bar code.

Because "<GS>" is not compatible with computer emulation data streams, the Uniform Code Council has been asked to change the specification. This change is made to send the same three character sequence "]C1" to identify the embedded FUNC1 codes.

This implementation should provide for clean application coding by identifying the same sequences for the same scanned codes. If the communication of Norand bar code types is enabled, the Shipping Container Label codes precede with a "J". These strings will appear on the computer display. The application may have to allow for strings longer than 48 characters (maximum length indicated in the specification). Actual length variance depends on the number of variable-length data fields. Allowing for 60 characters should be sufficient. Within the Code 128 specification, the computer can link bar codes together. If this is to happen, allow for more characters (computer limit is 100 characters).

The Application Identifier Standard, that is part of the UCC/EAN Shipping Label concept, complements, rather than replaces, other UCC/EAN standards. Most UCC/EAN standards primarily identify products.

Several industries expressed the need to standardize more than product identification. The UCC/EAN Code 128 Application Identifier Standard supplies this tool. The standard adds versatility for inter-enterprise exchanges of perishability dating, lot and batch identification, units of use measure, location codes, and several other information attributes.

For more detailed information on Code 128 UCC/EAN Shipping Label bar code and Application Identifier Standard, refer to the UCC/EAN-128 Application Identifier Standard specification.

I 2 of 5 (Interleaved)

I 2 of 5 (Interleaved 2 of 5 Code) is an all-numeric symbology, widely used for warehouse and heavy industrial applications. Its use has been particularly prevalent in the automobile industry. The I 2 of 5 symbology can be placed on smaller labels than what the standard UPC symbology requires.

I 2 of 5 also provides a little more flexibility on the type of material it can print on. Interleaved 2 of 5 Code has its name because of the way the bar code is configured.

I 2 of 5 bars and spaces both carry information. The bars represent the odd number position digits, while spaces represent the even number position digits. The two characters are interleaved as one. Messages encoded with this symbology have to use an even number of characters since two numeric characters always get interleaved together.

S 2 of 5 (Standard 2 of 5)

The code S 2 of 5 (Standard 2 of 5 Code) is designed primarily for:

- Warehouse inventory handling
- Identification of photo finishing envelopes
- Airline tickets
- Baggage and cargo handling

The code S 2 of 5 is simple and straightforward. All information is contained in the widths of the bars, with the spaces serving only to separate the individual bars.

Bars can either be wide or narrow, and the wide bars are usually three times the widths of the narrow bars. Spaces may be any reasonable width but are typically equal to the narrow bars. Narrow bars are identified as zero bits and wide bars as one bits.

Remember the code structure by associating the bar positions from left to right with weighting factors 1, 2, 4, 7, and parity. Exceptions to this rule are zero, start, and stop. This code is a discrete code, since the white spaces between the characters are not part of the code. Because the white spaces carry no information, their dimensions are not critical.

The S 2 of 5 code is self-checking, meaning a scanner passing through a printing void would detect the proper ratio of wide bars to total bars. When the scanner spots an error, a non-read will occur.

Plessey

Plessey finds its origin in the pulse width modulated (PWM) code developed in England. It is widely used for shelf markings in grocery stores. Pulse width modulated codes represent each bit of information by a bar and space pair. A zero bit consists of a narrow bar followed by a wide space, while a one bit consists of a wide bar followed by a narrow space. It is mainly a numeric symbology (0-9) with six extra characters available for assigning any symbol or letter desired.

Plessey codes are not self-checking and employ a variety of check characters. Plessey employs a polynomial-based Cyclic Redundancy Check (CRC). For start and stop characters, Plessey employs a 1101 and previously used a 0101.

This symbology is very limited about what information can be encoded. It is not considered for new applications.

MSI Code (Variant of Plessey)

In addition to Plessey characteristics, the MSI Code employs a Modulus 10 Check. For start and stop checks, MSI employs a single bit pair of 1 as a start symbol and a single bit pair of 0 as a stop symbol. MSI reverses the 1-2-4-8 BCD pattern for bit pair weighting to 8-6-2-1.

Bar Code Labels

You can change some settings on your 700 Series Computer by scanning the following Code 39 bar code labels.

- You can use the Unit Manager application to set the Automatic Shutoff, Volume, Backlight Timer, or Key Clicks parameters *(starting on page 352)*.
- You can use the Unit Manager application or the Data Collection control panel to set the three Virtual Wedge parameters (*starting on page* 325).



Note: When you use a bar code creation utility to make a scannable bar code label, the utility probably adds opening and closing asterisks automatically. Asterisks are included here for translation purposes.

Audio Volume



Note: The Audio Volume parameter information is on page 356.



\$+BV0

Set Audio Volume to very quiet

\$+VB1



\$+BV2

Set Audio Volume to normal (*default*)

\$+BV3



\$+BV4

Set Audio Volume to very loud

\$+BV5

Automatic Shutoff



Note: The Automatic Shutoff parameter information is on page 355.



\$+EZ1

Set Automatic Shutoff to 2 minutes

\$+EZ2

Set Automatic Shutoff to 3 minutes (default)

\$+EZ3

Set Automatic Shutoff to 4 minutes

\$+EZ4

Set Automatic Shutoff to 5 minutes

\$+EZ5

Backlight Timeout



Note: The Backlight Timeout parameter information is on page 353.



\$+DF10

Backlight Timeout 30 seconds

\$+DF30

Backlight Timeout 1 minute (default)

\$+DF60

Backlight Timeout 2 minutes

\$+DF120

Backlight Timeout 3 minutes

\$+DF180

Backlight Timeout 4 minutes

\$+DF240



\$+DF300

Key Clicks



Note: The Key Clicks parameter information is on page 354.



\$+KC0

Enable soft key clicks

\$+KC1

Enable loud key clicks (default)

\$+KC2

Virtual Wedge Grid, Preamble, Postamble

The following parameters are user-configurable strings. Refer to a full ASCII chart for more information.

Grid

For Virtual Wedge Grid, the first part of the bar code would be the following, which can include a string of up to 240 characters. *Parameter information starts on page 328.*

*\$+AF

Preamble

For Virtual Wedge Preamble, the first part of the bar code would be below, followed by a string of up to 31 characters (*no* <*NUL*>) and an asterisk. *Default is no characters. Parameter information is on page 326.*



*\$+AD

Postamble

For Virtual Wedge Postamble, the first part of the bar code would be below, followed by a string of up to 31 characters (*no* <*NUL*>) and an asterisk. *Default is no characters. Parameter information is on page 327.*



*\$+AE

Appendix C — Bar Codes



The Classes and Functions Index covers classes and functions for the 700 Series Color Mobile Computer.

The General Index covers all topics. Those in italics are figures, those in bold are tables.

The Files Index is to assist you in locating descriptions for device drivers, applications, utilities, batch files, or other files within this publication.

Classes and Functions

A

add_registry_section, [AddReg] flags, 244 registry_root_string, 244 value_name, 244 AddReg, [DefaultInstall], 240 [AddReg], add_registry_section flags, 244 registry_root_string, 244 value_name, 244 AddWep(), 103 AppName, [CEStrings], 237

B

BuildMax, [CEDevice], 238 BuildMin, [CEDevice], 238

С

CancelReadImage, IImage, 226 CancelReadRequest IADC, 152 IBarCodeReaderControl, 160 [CEDevice] BuildMax, 238 BuildMin, 238 ProcessorType, 238 UnsupportedPlatforms, 238 VersionMax, 238 VersionMin, 238 CESelfRegister, [DefaultInstall], 240 CESetupDLL, [DefaultInstall], 240 CEShortcuts, [DefaultInstall], 240 [CEShortcuts], shortcut list section shortcut filename, 245 shortcut_type_flag, 245 target_file/path, 245 target_file_path, 245 CESignature [SourceDiskNames], 240 [Version], 236 [CEStrings] AppName, 237 InstallDir, 237 Close, IImage, 228 CloseHandle() DTR printing, 134, 135 IrDA printing, 128 NPCP printing, 129, 130 ConfigureProfile(), 106 ControlLED, IBarCodeReaderControl, 161 Copyfiles, [DefaultInstall], 240 [CopyFiles], file list section destination_filename, 243 flags, 243 source_filename, 243 create/delete ADC COM objects, 149 CreateEvent(), 283

CreateFile() DTR printing, 134, 135 IrDA printing, 128 NPCP printing, 129, 130

D

[DefaultInstall] AddReg, 240 CESelfRegister, 240 CESetupDLL, 240 CEShortcuts, 240 Copyfiles, 240 DeregisterDevice(), 129 DTR printing, 134 [DestinationDirs], file_list_section, 242 DeviceIOControl(), 98 DTR printing, 134 NPCP printing, 129 DeviceIoControl(), NPCP printing, 130, 131 disk_ordinal, [SourceDiskNames], 240 DllRegisterServer, 240 DllUnregisterServer, 240

Ε

EnableWep(), 105 EncryptWepKeyForRegistry(), 106

F

file_list_section [CopyFiles] destination_filename, 243 flags, 243 source_filename, 243 [DestinationDirs], 242 filename, [SourceDiskFiles], 241

G

GetAssociationStatus(), 104 GetAuthenticationMode(), 104 GetBSSID(), 101 GetCodabar, IS9CConfig, 173 GetCode11, IS9CConfig2, 205 GetCode128, IS9CConfig, 180 GetCode39, IS9CConfig, 176, 200 GetCode93, IS9CConfig, 179 GetConfig, ISCP, 217 GetCustomSymIds, IS9CConfig2, 207 GetGlobalAmble, IS9CConfig2, 211 GetI2of5, IS9CConfig, 183 GetLinkSpeed(), 102 GetMac(), 101 GetMatrix2of5, IS9CConfig, 185 GetMSI, IS9CConfig, 187 GetNetworkMode(), 103 GetNetworkType(), 102 GetPDF417, IS9CConfig, 188 GetPDF417Ext, IS9CConfig2, 213 GetPlessey, IS9CConfig, 192 GetPowerMode(), 105 GetRSSI(), 103 GetRTSThreshold(), 106 GetSSID(), 101

GetStandard2of5, IS9CConfig, 194 GetSymIdXmit, IS9CConfig2, 214 GetTelepen, IS9CConfig, 197 GetTXPower(), 102 GetUpcEan, IS9CConfig, 198 GetWepStatus(), 104

I

IADC, 151 CancelReadRequest, 152 Initialize, 153 QueryAttribute, 154 QueryData, 155 Read, 156 SetAttribute, 157 IBARCODEREADER.H, IBarCodeReaderControl functions, 159 IBarCodeReaderControl, 159 CancelReadRequest, 160 ControlLED, 161 Initialize, 162 IssueBeep, 163 QueryAttribute, 164 Read, 165 SetAttribute, 167 TriggerScanner, 171 IImage CancelReadImage, 226 Close, 228 Open, 227 ReadImage, 225 ReadSigCapBuffer, 221 ReadSigCapFile, 224 Start, 226 Stop, 227 Imager, settings, IS9CConfig3, 216 Initialize IADC, 153 IBarCodeReaderControl, 162 InstallDir, [CEStrings], 237 IS9CConfig, 172 GetCodabar, 173 GetCode128, 180 GetCode39, 176, 200 GetCode93, 179 GetI2of5, 183 GetMatrix2of5, 185 GetMSI, 187 GetPDF417, 188 GetPlessey, 192 GetStandard2of5, 194 GetTelepen, 197 GetUpcEan, 198 SetCodabar, 174 SetCode128, 181 SetCode39, 177 SetCode93, 179 SetI2of5, 184 SetMatrix2of5, 186

SetMSI, 187 SetPDF417, 189 SetPlessey, 192 SetStandard2of5, 195 SetTelepen, 197 IS9CConfig2, 204 GetCode11, 205 GetCustomSymIds, 207 GetGlobalAmble, 211 GetPDF417Ext, 213 GetSymIdXmit, 214 SetCode11, 205 SetCustomSymIds, 208 SetGlobalAmble, 212 SetPDF417Ext, 213 SetSymIdXmit, 214 IS9CConfig3, 216 ISCP GetConfig, 217 SetConfig, 218 isOrinoco(), 106 IssueBeep, IBarCodeReaderControl, 163 ITCDeviceClose, 150 ITCDeviceOpen, 140, 149 ITCUUID.LIB, IBarCodeReaderControl functions, 159

Κ

KernelIoControl(), 264

0

Open, IImage, 227 OSVERSIONINFO.dwBuildNumber, 238 OSVERSIONINFO.dwVersionMajor, 238 OSVERSIONINFO.dwVersionMinor, 238

Ρ

ProcessorType, [CEDevice], 238 Provider, [Version], 236

Q

QueryAttribute IADC, 154 IBarCodeReaderControl, 164 QueryData, IADC, 155

R

RadioConnect(), 101 RadioDisconnect(), 101 Read, 140 IADC, 156 IBarCodeReaderControl, 165 ReadFile(), NPCP printing, 129 ReadImage, IImage, 225 ReadSigCapBuffer, IImage, 221 ReadSigCapFile, IImage, 224 RegFlushKey(), 81, 262, 270 RegisterDevice(), 129 DTR printing, 134 RegOpenKeyEx(), 282 RegQueryValueEx(), 282

S

SetAttribute IADC, 157 IBarCodeReaderControl, 167 SetAuthenticationMode(), 104 SetChannel(), 105 SetCodabar, IS9CConfig, 174 SetCode11, IS9CConfig2, 205 SetCode128, IS9CConfig, 181 SetCode39, IS9CConfig, 177 SetCode93, IS9CConfig, 179 SetConfig, ISCP, 218 SetCustomSymIds, IS9CConfig2, 208 SetGlobalAmble, IS9CConfig2, 212 SetI2of5, IS9CConfig, 184 SetMatrix2of5, IS9CConfig, 186 SetMSI, IS9CConfig, 187 SetNetworkMode(), 103 SetPDF417, IS9CConfig, 189 SetPDF417Ext, IS9CConfig2, 213 SetPlessey, IS9CConfig, 192 SetRTSThreshold(), 106 SetSSID(), 105 SetStandard2of5, IS9CConfig, 195 SetSymIdXmit, IS9CConfig2, 214 SetTelepen, IS9CConfig, 197 SHFullScreen(), 262, 263 shortcut_list_section, [CEShortcuts] shortcut_filename, 245 shortcut_type_flag, 245 target_file/path, 245 target_file_path, 245

Signature, [Version], 236 [SourceDiskFiles], filename, 241 [SourceDiskNames] CESignature, 240 disk_ordinal, 240 SourceDisksNames.MIPS, 241 SourceDisksNames.SH3, 241 Start, IImage, 226 StartScanList(), 107 Stop, IImage, 227 string_key, [Strings], 237 [Strings], string_key, 237 SYSTEMINFO.dwProcessorType, 238

Т

Trigger settings, IS9CConfig3, 216 TriggerScanner, IBarCodeReaderControl, 171

U

UnsupportedPlatforms, [CEDevice], 238

V

[Version] CESignature, 236 Provider, 236 Signature, 236 VersionMax, [CEDevice], 238 VersionMin, [CEDevice], 238

W

WriteFile() DTR printing, 134, 135 IrDA printing, 128 NPCP printing, 129, 130

General Index

Numbers

1470 Imager. See Imager 1551/1553 Tethered Scanner. See Tethered scanner 1D laser scanner, about, 137 2D Imager about, 137 data collection features, 146 aimer LED, 146 scaled illumination LED, 146 window size and position, 146 image acquisition features, 147 overview, 146 4820 printer, NPCP driver, 129 6804DM printer DTR driver, 134 IrDA driver, 128 6804T printer DTR driver, 134 IrDA driver, 128 6805A printer DTR driver, 134 IrDA driver, 128 6806 printer DTR driver, 134 IrDA driver, 128 6808 printer DTR driver, 134 IrDA driver, 128 printer support, 127 681T printer, DTR driver, 134 6820 printer IrDA driver, 128 NPCP driver, 129 printer support, 127 6920 Communications Server, ManifestName parameter, 254 700 Platform Build, version number, 342 740 Color Computer, 281 781 printers DTR driver, 134 printer support, 127 782T printer, printer support, 127 802.11 CR radio CORE module, 107 802.11 WEP Encryption, profile security information, 91 802.11b antenna color code, 85 API, 100 channel, 89 communications setup, 87 configuration profiles, 100 CORE module, 107 network type, 89 profiles, 87 basic information, 89 certificates, 95 exporting, 96

import/export, 96 importing, 97 read-only, 94 scan list, 97, 98 security information, 90 selected, 97 SSID (network name), 89 WEP encryption, 91 802.1x TLS, profile security information, 92 802.1x TTLS, profile security information, 93

A

Abstract Syntax Notation.1. See ASN.1 ActiveSync ActiveSync Help, 30 adding programs, 26 adding programs to Start menu, 28 Folder behavior connected to e-mail server, 46 installing applications, 77 Microsoft Reader, 58 Pocket Internet Explorer favorite links, 62 Mobile Favorites folder, 62 Pocket PC, 29 Pocket PC icon, 13 Pocket PC status icons, 12 URL, 29 ActiveX control tools, unit information control panel, CAB files, 345 AD command, with/without data, 326 ADC COM interfaces, 138 functions create/delete objects, 149 IADC, 151 IBarCodeReaderControl, 159 IS9CConfig, 172 IS9CConfig2, 204 IS9CConfig3, 216 Adding a profile, 88 Adding bookmarks, Microsoft Reader, 61 Adding drawings to text, Microsoft Reader, 61 Adding programs ActiveSync, 26 Pocket Internet Explorer, 27 Pocket PC, 26 to the Start menu, 28 via ActiveSync, 28 via File Explorer, 28 Adjusting settings, Pocket PC, 26 Adobe Acrobat Reader, URL, 116 AE command, with/without data, 327 Aimer LED duration, configuration parameter, 323 Alpha plane on keypad, 281 Annotations index, Microsoft Reader, 61 Antenna, radio type, 85 APIs 802.11b, 100 AT command interface, 115 IrSock, 128 Appointments, via Calendar, 31

APS linear imager, about, 137 ASCII printing, 128 printing to a port, port print method, 128 raw text to printer, 128 ASN.1, 125 Asset management, DeviceURL parameter, 253 AT command interface, 115 terminal application, 114 testing, 117 Attaching notes to text, Microsoft Reader, 61 Audio files, Windows Media Player, 57 Audio system external headset jack, 2 internal microphone, 2 speaker, 2 AutoCab, command line syntax, 82 AutoFTP, 260 AutoIP, 122 Automatic Data Collection. See ADC COM interfaces Automatic Private IP. See AutoIP Automatic shutoff bar code configuration, 369 configuration parameter, 355 Autostart FTP, 260 AvantGo channels, Pocket Internet Explorer, 64 AXCommunication, 345 AXFileTransfer, 345 AXReaderCommand, 345 AXVWedge, 345

B

Backlight timeout bar code configuration, 369 configuration parameter, 353 Bar Code scanning labels, 368 supported symbologies, 147, 289 symbologies, 360 Codabar, 362 Code 11, 363 Code 128, 364 Code 39, 363 Code 39 concatenation, 363 Code 39 full ASCII, 363 Code 93, 364 data string formats, 360 EAN, 362 I 2 of 5, 366 MSI code, 367 Plessey, 367 S 2 of 5, 366 UPC, 362 Bar code configuration audio volume, 368 automatic shutoff, 369

backlight timeout, 369 Code 39, 368 key clicks, 370 BARCODE_DATA_TYPE_ASCII, IBarCodeReaderControl::Read, 165 BARCODE_DATA_TYPE_UNICODE, IBarCodeReader-Control::Read, 165 BARCODE_DATA_TYPE_UNKNOWN, IBarCodeReaderControl::Read, 165 Battery low battery conditions, 3 Pocket PC status icons, 12 status, 2 Battery status, unit information control panel applet, 343 Beeper configuration parameter frequency, 320 volume, 318 silencing the volume, 319 supported functions, 317 Unit Manager, 349 when not available beeper frequency, 320 good read beep duration, 322 good read beeps, 321 biActualImageSize, pImgBuffer, IImage::ReadSigCapBuffer, 2.2.2 biBitCount, pImgBuffer, IImage::ReadSigCapBuffer, 222 biHeight, pImgBuffer, IImage::ReadSigCapBuffer, 222 biMaxImageBytes, pImgBuffer, IImage::ReadSigCapBuffer, 222 biWidth, pImgBuffer, IImage::ReadSigCapBuffer, 222 Block recognizer, Pocket PC input panel, 17 BlockSize, FTP Server, 252 Bluealps CORE module, 120 Bluetooth CORE module, 120 unit information control panel, main stack CAB file, 344 Bluetooth compatibility, network support, 120 Bluetooth Device Manager, documentation, 120 Books, Microsoft Reader adding bookmarks, 61 adding drawings, 61 annotations index, 61 attaching notes, 61 copying, 61 downloading, 58 highlighting, 61 reading, 60 removing, 61 searching, 61 Browing the Internet, Pocket Internet Explorer, 66 BTctrl program, documentation, 120 Build information, 5 byFNC1, IS9CConfig::SetCode128, 181

С

CAB files after the extraction, 248 creating, 236 INF files, 236 with CAB Wizard, 249 information regarding, 4 installation functions, SETUP.DLL, 248 placing files onto storage card, 80 unit information control panel applet, 344 Cabinet Wizard creating CAB files, 249 troubleshooting, 250 using the application, 236 Cabling, scanner, 232 Calendar creating an appointment, 32 meeting requests, 33 Pocket Outlook, 31 Pocket PC icon, 13 scheduling a meeting, 33 using the summary screen, 33 Capacitor, internal super, 3 Capturing thoughts and ideas, via Notes, 40 Card support CompactFlash cards, 5 modems, 4 MultiMediaCards, 5 radios, 4 SecureDigital cards, 5 CDMA/1xRTT, 110 antenna color code, 85 AT command set, 116 CORE module, 111 CEImager location of the executable file, 80 migrating AUTORUN.DAT files, 80 Channel, 802.11 radio module, 89 ClassID field values VN CLASS ASIC, 266 VN_CLASS_BOOTSTRAP, 266 VN_CLASS_KBD, 266 Clock Pocket PC settings, 26 setting date and time, 358 Closing drivers, NPCP, 130 CMIP, 123 Codabar, 362 configuration parameter, 292 user ID, 310 default S9C settings, 175 enumerations, 175 IS9CConfig::GetCodabar, 173 IS9CConfig::SetCodabar, 174 modifier characters, 219 Code 11, 363 configuration parameter, 306 user ID, 314

default S9C settings, 206 enumerations, 206 IS9CConfig2::GetCode11, 205 IS9CConfig2::SetCode11, 205 modifier characters, 219 Code 128, 364 configuration parameter, 295 FNC1 character, 297 user ID, 310 default S9C settings, 181 enumerations, 182 IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 IS9CConfig::GetCode128, 180 IS9CConfig::SetCode128, 181 modifier characters, 219 Code 39, 363 configuration parameter, 290 user ID, 310 default S9C settings, 177 enumerations, 178 IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 IS9CConfig::GetCode39, 176 IS9CConfig::SetCode39, 177 modifier characters, 219 Code 93, 364 configuration parameter, 294 length, 294 user ID, 311 default S9C settings, 179 enumerations, 180 IS9CConfig::GetCode93, 179 IS9CConfig::SetCode93, 179 modifier characters, 219 Code Division Multiple Access. See CDMA/1xRTT Codes 11, 363 128, 364 39, 363 39 concatenation, 363 39 full ASCII, 363 93, 364 Cold boot, IOCTL_HAL_COLDBOOT, 275 COM port configuration, 231 wedge settings, 231 COM1, NPCP parameter, 129 COM1 port, 128 Comm port wedge disabling, 231 enabling, 230 error messages, 232 limitations, 233 settings, 231 unit information control panel, 344 Command line syntax, AutoCab, 82 Common Object Resource Environment. See CORE Communications DTR, 135 NPCP, 132 Communications options, 83 CompactFlash cards card support, 5 installing applications, 78 Composing Messages, via Inbox, 45 Computer shutdown, 3 Concatenation, 363 Configuration Management. See Unit Manager Configuration parameters aimer LED duration, 323 automatic shutoff, 355 backlight timeout, 353 beeper frequency, 320 volume, 318 codabar, 292 user ID, 310 code 11, 306 user ID, 314 code 128, 295 FNC1 character, 297 user ID, 310 code 39, 290 user ID, 310 code 93, 294 length, 294 user ID, 311 data matrix, 308 date/time, 352 EAN 13 user ID, 313 8 user ID, 313 good read beep duration, 322 beeps, 321 identification contact, 338 location, 340 name, 339 image dimension, 324 interleaved 2 of 5, 303 user ID, 311 key clicks, 354 macro PDF, 300 matrix 2 of 5, 304 user ID, 313 micro PDF 417, 302 MSI, 299 user ID, 311 PDF 417, 300 user ID, 311 plessey, 298 user ID, 312 prefix, 315

security encryption key, 335 read encryption, 333 read-only community string, 331 read/write community string, 332 write encryption, 334 SNMP, security subnet mask, 309 standard 2 of 5, 291 user ID, 312 suffix, 316 telepen, 305 user ID, 313 trap authentication, 336 threshold, 337 UPC A user ID, 312 E user ID, 312 **UPC/EAN**, 293 virtual wedge, 325 code page, 329 grid, 328 postamble, 327 preamble, 326 volume, 356 Connecting directly to e-mail server, via Inbox, 42 Connecting to an ISP, 68 e-mail server, 72 work, 70 Connections See also Getting connected directly to e-mail server, 72 ending, 72 setting up an e-mail service, 73 status icon, 12 to an ISP, 68 via Ethernet, 69 via modem, 68 to work, 70 via Ethernet, 71 via modem, 70 via Ethernet to an ISP. 69 to work, 71 via modem to an ISP, 68 to work, 70 Contacts creating a contact, 34 finding a contact, 35 MSN Messenger chatting with, 55 working with, 54 Pocket Outlook, 34 Pocket PC icon, 13 using the summary screen, 36

QR code, 307

Control panel applets clock, 358 data collection, 288 beeper/LED, 317 imager, 323 symbologies, 289 symbology options, 309 virtual wedge, 325 power, battery status, 2 SNMP, 330 identification, 338 security, 331 traps, 336 system, wireless network, 87 unit information, 341 battery status, 3, 343 CAB files, 344 versions, 6, 342 utilities, 81 Converting writing to text, 20 Copying text, Microsoft Reader, 61 **CORE**, 84 802.11b radio module, 107 details, 109 general, 107 accessing from Programs panel, 84 Today screen, 84 Bluealps module, 120 Bluetooth, 120 module for 802.11b NIC, 100 WAN radio module, 111 details, 113 general, 111 Creating a modem connection to an ISP, 68 to work, 70 an Ethernet connection to an ISP, 69 to work, 71 appointment via Calendar, 32 CAB files, 236 with CAB Wizard, 249 contact via Contacts, 34 document via Pocket Word, 47 drawing via Notes, 22 INF files, 236 meeting requests, 33 note via Notes, 41 recording via Notes, 23 task via Tasks, 38 workbook via Pocket Excel, 52 Customer Support, xviii D Data collection 2D imager features, 146 ADC COM interfaces, 138

configuration parameters aimer LED duration, 323 beeper frequency, 320 beeper volume, 318 codabar, 292 codabar user ID, 310 code 11, 306 code 11 user ID, 314 code 128, 295 code 128 FNC1 character, 297 code 128 user ID, 310 code 39, 290 code 39 user ID, 310 code 93, 294 code 93 length, 294 code 93 user ID, 311 data matrix, 308 EAN-13 user ID, 313 EAN-8 user ID, 313 good read beep duration, 322 good read beeps, 321 image dimension, 324 interleaved 2 of 5, 303 interleaved 2 of 5 user ID, 311 macro PDF, 300 matrix 2 of 5, 304 matrix 2 of 5 user ID, 313 micro PDF 417, 302 MSI, 299 MSI user ID, 311 PDF 417, 300 PDF 417 user ID, 311 plessey, 298 plessey user ID, 312 prefix, 315 OR code, 307 standard 2 of 5, 291 standard 2 of 5 user ID, 312 suffix, 316 telepen, 305 telepen user ID, 313 UPC-E user ID, 312 UPC-A user ID, 312 **UPC/EAN**, 293 virtual wedge, 325 virtual wedge code page, 329 virtual wedge grid, 328 virtual wedge postamble, 327 virtual wedge preamble, 326 functions create/delete ADC COM objects, 149 IADC, 151 IBarCodeReaderControl, 159 IS9CConfig, 172 IS9CConfig2, 204 IS9CConfig3, 216 initialization, 138 Unit Manager, 348 Data filtering, virtual wedge grid, 141

build version number, 342

Data Matrix configuration parameter, 308 IS9CConfig3 function, 216 Date, setting, 358 Date/Time, configuration parameter, 352 Deleting a profile, 88 DeviceName, FTP Server, 253 DeviceURL, FTP Server, 253 DHCP, 122 Display full screen, 263 Docks, modem support, 4 DRAM, low battery shutdown, 3 Drawing mode, Pocket Word, 51 Drawing on the screen See also Notes Pocket Word, 51 Drivers DTR communications, 135 installing, 134 opening, 135 removing, 134 writing to, 135 NPCP closing, 130 communications, 132 I/O controls, 131 installing, 129 opening, 130 reading from, 130 removing, 129 writing to, 130 O'Neil. See DTR printing DTR printing, 134 closing driver, 135 communications, 135 opening driver, 135 removing driver, 134 writing to driver, 135 dwAttrBufferSize IBarCodeReaderControl::QueryAttribute, 164 IBarCodeReaderControl::SetAttribute, 169 dwBufferSize IADC::QueryAttribute, 154 IS9CConfig2::GetGlobalAmble, 211 IS9CConfig2::SetGlobalAmble, 212 dwCommandBuffSize ISCP::GetConfig, 217 ISCP::SetConfig, 218 dwDataBufferSize IADC::Read, 156 IBarCodeReaderControl::Read, 165 dwLength IS9CConfig::SetCode128, 181 IS9CConfig::SetCode39, 177 IS9CConfig::SetCode93, 179 IS9CConfig::SetMatrix2of5, 186 IS9CConfig::SetMSI, 187 IS9CConfig::SetPlessey, 192

dwMaxNumElement, IS9CConfig2::GetCustomSymIds, 207dwNextMessageSize, IADC::QueryData, 155 dwNumberOfBeeps, IBarCodeReaderControl::IssueBeep, 163 dwNumBytes IS9CConfig::SetCodabar, 174 IS9CConfig::SetI2of5, 184 IS9CConfig::SetStandard2of5, 195 dwNumElement, IS9CConfig2::SetCustomSymIds, 208 dwReplyBuffMaxSize ISCP::GetConfig, 217 ISCP::SetConfig, 218 dwStructSize pImgBuffer, IImage::ReadSigCapBuffer, 222 pSigCapSpec, IImage::ReadSigCapBuffer, 221 dwTimeout IADC::Read, 156 IBarCodeReaderControl::Read, 166 IImage::ReadImage, 225 dwTotalBufferedBytes, IADC::QueryData, 155

Е

E-mail server getting connected, 72 setting up a service, 73 eAmbleId IS9CConfig2::GetGlobalAmble, 211 IS9CConfig2::SetGlobalAmble, 212 EAN, 362 configuration parameter, 293 13 user ID, 313 8 user ID, 313 default S9C settings, 181, 201 enumerations, 201 IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 modifier characters, 220 ean13Check IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 ean13Select IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 ean8Check IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 ean8Reencode IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 ean8Select IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 eAttr IBarCodeReaderControl::QueryAttribute, 164 IBarCodeReaderControl::SetAttribute, 167 eAttribID IADC::QueryAttribute, 154 IADC::SetAttribute, 157

eCheck IS9CConfig::SetCodabar, 174 IS9CConfig::SetI2of5, 184 IS9CConfig::SetMSI, 187 IS9CConfig::SetPlessey, 192 IS9CConfig::SetStandard2of5, 195 IS9CConfig2::SetCode11, 205 eCip128State, IS9CConfig::SetCode128, 181 eCLSI, IS9CConfig::SetCodabar, 174 eCode128, IS9CConfig2::SetPDF417Ext, 213 eDataType, IBarCodeReaderControl::Read, 165 eDecode IS9CConfig::SetCodabar, 174 IS9CConfig::SetCode128, 181 IS9CConfig::SetCode39, 177 IS9CConfig::SetCode93, 179 IS9CConfig::SetI2of5, 184 IS9CConfig::SetMatrix2of5, 186 IS9CConfig::SetMSI, 187 IS9CConfig::SetPlessey, 192 IS9CConfig::SetStandard2of5, 195 IS9CConfig::SetTelepen, 197 IS9CConfig2::SetCode11, 205 IS9CConfig2::SetPDF417Ext, 213 eDepth, pSigCapSpec, IImage::ReadSigCapBuffer, 222 eDeviceFlags IADC::Initialize, 153 IBarCodeReaderControl::Initialize, 162 ITCDeviceOpen, 149 Editing a profile, 88 Edition information, 8 eEan128Ident, IS9CConfig::SetCode128, 181 eFormat IImage::ReadImage, 225 IS9CConfig::SetCode39, 177 IS9CConfig::SetStandard2of5, 195 IS9CConfig::SetTelepen, 197 pImgBuffer, IImage::ReadSigCapBuffer, 222 pSigCapSpec, IImage::ReadSigCapBuffer, 222 eLED, IBarCodeReaderControl::ControlLED, 161 eLengthId IS9CConfig::SetCodabar, 174 IS9CConfig::SetI2of5, 184 IS9CConfig::SetStandard2of5, 195 eMacroPdf, IS9CConfig::SetPDF417, 189 Embedded modules, SB555, 110 Encoded Code 39 concatenation, 363 full ASCII, 363 Ending a connection, 72 Enumerations Codabar, 175 Code 11, 206 Code 128, 182 Code 39, 178 Code 93, 180 Interleaved 2 of 5, 185 Matrix 2 of 5, 186 MSI, 188 PDF 417, 190

Plessey, 193 Standard 2 of 5, 196 Telepen, 198 UPC/EAN, 201 ePdf417Decode, IS9CConfig::SetPDF417, 189 ePdfAddressee, IS9CConfig::SetPDF417, 189 ePdfChecksum, IS9CConfig::SetPDF417, 189 ePdfControlHeader, IS9CConfig::SetPDF417, 189 ePdfFileName, IS9CConfig::SetPDF417, 189 ePdfFileSize, IS9CConfig::SetPDF417, 189 ePdfSegmentCount, IS9CConfig::SetPDF417, 189 ePdfSender, IS9CConfig::SetPDF417, 189 ePdfTimeStamp, IS9CConfig::SetPDF417, 189 Epson Escape Sequences, 128 Error messages comm port wedge, 232 tethered scanner, 232 ERROR_INSUFFICIENT_BUFFER IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 ERROR_INVALID_PARAMETER IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 eSS IS9CConfig::SetCodabar, 174 IS9CConfig::SetCode39, 177 eSSChars, IS9CConfig::SetCode39, 177 eSymbology, IBarCodeReaderControl::Read, 165 eSymIdXmit, IS9CConfig2::SetSymIdXmit, 214 Ethernet communications setup, 86 creating a connection to an ISP, 69 to work, 71 ETSI GSM 07.05 interface specifications, 115 ETSI GSM 07.07 interface specifications, 115 European Article Numbering code. See EAN eVer, IS9CConfig2::SetCode11, 205 Excel. See Pocket Excel Exporting a profile, 802.11 radio module, 96

F

Factory repair, xviii Favorite links, Pocket Internet Explorer, 62 File Explorer adding programs to Start menu, 28 Pocket PC, 25 removing programs, 28 File Transfer Protocol. See FTP Filter expression values, virtual wedge grid, 142 Find feature, Pocket PC, 25 fLedOn, IBarCodeReaderControl::ControlLED, 161 FlushBufferedData IADC::CancelReadRequest, 152 IBarCodeReaderControl::CancelReadRequest, 160 Folder behavior connected to e-mail server ActiveSync, 46 IMAP4, 46 POP3, 46 SMS, 46

FRAME_NOT_ACKED, 131 fScannerOn, IBarCodeReaderControl::TriggerScanner, 171 fSigCapEnable, IImage::Open, 227 FTP client, 256 configurable parameters, 252 BlockSize, 252 DeviceName, 253 DeviceURL, 253 IDNATarget, 254 ManifestName, 254 PauseAtStartup, 255 Root, 255 FTPDCMDS subdirectory, 259 heartbeat, 256 RTC 959, 259 server, 256 installing applications, 78 server requests CDUP, 256 CWD, 256 **DELE**, 256 HELP, 256 LIST, 256 MKD, 256 MODE, 256 NLST, 256 NOOP, 256 PASS, 256 PWD, 256 **OUIT**, 256 **RETR**, 256 RMD, 256 RNFR, 256 RNTO, 257 SITE, 257 SITE ATTRIB, 257 SITE BOOT, 258 SITE COPY, 258 SITE EXIT, 258 SITE HELP, 258 SITE KILL, 258 SITE LOG, 258 SITE PLIST, 258 SITE RUN, 258 SITE STATUS, 259 SITE TIMEOUT, 259 STOR, 257 SYST, 257 **TYPE**, 257 USER, 257 XCUP, 257 XCWD, 257 XMKD, 257 XPWD, 257 XRMD, 257 stopping server from application, 260

support, 256 web browsers, 259 FTPDCMDS subdirectory, FTP support, 259 Full screen display, 263 G GDI approach, 128 General Packet Radio Service. See GSM/GPRS Getting connected directly to an e-mail server, 72 infrared (IR) port, 67 ISP, 67 Pocket PC, 67 setting up an e-mail service, 73 to an ISP, 68 creating a modem connection, 68 creating an Ethernet connection, 69 to work, 70 creating a modem connection, 70 creating an Ethernet connection, 71 transfer items using infrared, 67 Global services and support center, xviii Gold plane on keypad, 281 Good read, configuration parameter beep duration, 322 beeps, 321 Grid data configuration parameter, 328 filtering, 141 GSM/GPRS, 110 antenna color code, 85 AT command set GEM350X, 116 MC45, 116 CORE module, 111 н HAL, verion of Pocket PC IOCTL_HAL_GET_BOOTLOADER_VERINFO, IOCTL_HAL_GET_OAL_VERINFO, 273 Header files IADC.H, IADC functions, 151 IBARCODEREADER.H, IBarCodeReaderControl functions, 159 IS9CCONFIG.H IS9CConfig functions, 172 IS9CConfig2 functions, 204 ITCDEVMGMT.H, 149 Headset jack, external, 2 Highlighting text, Microsoft Reader, 61 Hotmail account, 53

I 2 of 5. *See* Interleaved 2 of 5 I/O controls, NPCP driver, 131 iAspectRatio, pSigCapSpec, IImage::ReadSigCapBuffer, 221

ID field values IOCTL_HAL_ITC_READ_PARM ITC NVPARM 80211 INSTALLED, 268 ITC_NVPARM_80211_RADIOTYPE, 268 ITC_NVPARM_ANTENNA_DIVERSITY, 267 ITC_NVPARM_BLUETOOTH_INSTALLED, 268 ITC_NVPARM_CONTRAST, 266 ITC_NVPARM_DISPLAY_TYPE, 266 ITC_NVPARM_ECN, 266 ITC_NVPARM_EDBG_SUBNET, 266 ITC_NVPARM_EDG_IP, 266 ITC_NVPARM_ETHERNET_ID, 265 ITC_NVPARM_INTERMEC_DATACOLLEC-**TION HW, 267** ITC NVPARM INTERMEC DATACOLLEC-**TION SW, 267** ITC NVPARM INTERMEC SOFTWARE CON-TENT, 267 ITC_NVPARM_LAN9000_INSTALLED, 269 ITC_NVPARM_MANF_DATE, 265 _NVPARM_MCODE, 266 ITC ITC NVPARM RTC RESTORE, 267 ITC_NVPARM_SERIAL_NUM, 265 ITC_NVPARM_SERIAL2_INSTALLED, 269 ITC_NVPARM_SERVICE_DATE, 266 ITC_NVPARM_SIM_PRO-TECT_HW_INSTALLED, 269 ITC NVPARM SIM PRO-TECT SW INSTALLED, 269 ITC NVPARM VERSION NUMBER, 266 ITC_NVPARM_VIBRATE_INSTALLED, 269 ITC_NVPARM_WAN_FREQUENCY, 268 ITC NVPARM WAN INSTALLED, 268 ITC_NVPARM_WAN_RADIOTYPE, 268 ITC_NVPARM_WAN_RI, 267 IOCTL_HAL_ITC_WRITE_SYSPARM ITC_DOCK_SWITCH, 270 ITC_WAKEUP_MASK, 271 ITC_AMBIENT_FRONTLIGHT, 271 ITC_AMBIENT_KEYBOARD, 271 ITC_REGISTRY_LOCATION, 270 ITC_REGISTRY_SAVE_ENABLE, 270 Identification configuration parameter contact, 338 location, 340 name, 339 Unit Manager, 351 **IDNA** DeviceName, 253 DeviceURL, 253 IDNATarget, 254 ManifestName, 254 IDNATarget, FTP Server, 254 iid, ITCDeviceOpen, 149 IImage interface, 221

Image, acquisition features, 147 Image dimension, configuration parameter, 324 Imager beeper functions not available beeper frequency, 320 good read beep duration, 322 good read beeps, 321 control panel appet, data collection, 323 data collection parameters aimer LED duration, 323 data matrix, 308 image dimension, 324 QR code, 307 settings, 232 supported beeper functions, 317 functions, 323 symbologies, 289 supported symbologies, 147 symbologies not available CIP 128 French Pharmaceutical, 296 Code 11, 306 Code 128 FNC1 character, 297 EAN 128]C1, 296 Macro PDF, 300 Matrix 2 of 5, 304 Micro PDF 417, 302 Telepen, 305 symbology user IDs not available Codabar, 310 Code 11, 314 Code 128, 310 Code 39, 310 Code 93, 311 EAN 13, 313 EAN 8, 313 Interleaved 2 of 5, 311 Matrix 2 of 5, 313 MSI, 311 PDF 417, 311 Plessey, 312 Standard 2 of 5, 312 Telepen, 313 UPC A, 312 UPC E, 312 Unit Manager, 350 IMAP4, Folder behavior connected to e-mail server, 46 Import libraries ITCDEVMGMT.LIB, 149 **ITCUUID.LIB** IADC functions, 151 IBarCodeReaderControl functions, 159 IS9CConfig functions, 172 IS9CConfig2 functions, 204 Importing a profile, 802.11 radio module, 97

Inbox composing messages, 45 connecting directly to e-mail server, 42 Folder behavior connected to e-mail server, 46 getting connected, 67 managing e-mail messages and folders, 46 Pocket Outlook, 42 Pocket PC icon, 13 synchronizing e-mail messages, 42 using a message list, 43 using My Text, 24 INF files, creating, 236 Infrared (IR) port Pocket PC, 67 transfer items using, 67 receiving information, 67 sending information, 67 Input panel block recognizer, 17 keyboard, 17 letter recognizer, 18 methods available, 16 Pocket PC, 14 Pocket Word, 49 selecting typed text, 18 transcriber, 18 word suggestions, 17 Installation, site, xviii Installation functions, SETUP.DLL, 248 Installing applications using a storage card, 78 using CompactFlash cards, 78 using SecureDigital cards, 79 with ActiveSync, 77 with Application Manager, 78 with FTP Server, 78 Installing drivers DTR, 134 NPCP, 129 Instant messaging, 53 Pocket PC icon, 12 Intelliget Bar Code Unit IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 Interface specifications, ETSI GSM 07.0x, 115 Interleaved 2 of 5, 366 configuration parameter, 303 user ID, 311 default S9C settings, 184 enumerations, 185 IS9CConfig::GetI2of5, 183 IS9CConfig::SetI2of5, 184 modifier characters, 219 Intermec Device Network Announcement. See IDNA Internet Explorer. See Pocket Internet Explorer Internet explorer Pocket PC 2002 edition, 8 software build version, 5 Internet Service Provider. See ISP IOCTL_GET_CPU_ID, 280

IOCTL_HAL_COLDBOOT, 275, 280 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274 IOCTL_HAL_GET_DEVICE_INFO, 264 IOCTL_HAL_GET_DEVICEID, 272 IOCTL_HAL_GET_OAL_VERINFO, 273 IOCTL_HAL_GET_RESET_INFO, 276 IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 IOCTL_HAL_REBOOT, 278, 280 IOCTL_HAL_WARMBOOT, 275, 280 IOCTL_LOAD_NDIS_MINIPORT, 98 IOCTL_NPCP_BIND, 131 IOCTL_NPCP_CANCEL, 131 IOCTL NPCP CLOSE, 131 IOCTL_NPCP_ERROR, 131 IOCTL_NPCP_FLUSH, 131 IOCTL_PROCESSOR_INFORMATION, 279 IOCTL_UNLOAD_NDIS_MINIPORT, 98 iOffsetX, pSigCapSpec, IImage::ReadSigCapBuffer, 221 iOffsetY, pSigCapSpec, IImage::ReadSigCapBuffer, 222 IrDA printing, 128 iResolution, pSigCapSpec, IImage::ReadSigCapBuffer, 222 IS9CConfig3 Data Matrix symbology, 216 imager settings, 216 QRCode symbology, 216 trigger settings, 216 ISP connecting to via Pocket PC, 68 creating a modem connection, 68 an Ethernet connection, 69 Pocket Internet Explorer, 62 Pocket PC, 67 ITC_DOCK_SWITCH, 270 ITC_ WAKEUP_MASK, 271 ITC_AMBIENT_FRONTLIGHT, 271 ITC_AMBIENT_KEYBOARD, 271 ITC_BARCODE_LASER_GOOD_READ_LED, IBar-CodeReaderControl::ControlLED, 161 ITC DEVID 80211RADIO INTEL 2011B, 268 ITC_DEVID_80211RADIO_MAX values ITC_DEVID_80211RADIO_INTEL_2011B, 268 ITC_DEVID_80211RADIO_NONE, 268 ITC_DEVID_80211RADIO_NONE, 268 ITC_DEVID_INTERMEC2D_IMAGER, 267 ITC DEVID OEM2D IMAGER, 267 ITC_DEVID_SCANHW_MAX values ITC_DEVID_INTERMEC2D_IMAGER, 267 ITC_DEVID_OEM2D_IMAGER, 267 ITC_DEVID_SCANHW_NONE, 267 ITC_DEVID_SE900_LASER, 267 ITC_DEVID_SE900HS_LASER, 267 ITC DEVID SCANHW NONE, 267 ITC DEVID SE900 LASER, 267 ITC_DEVID_SE900HS_LASER, 267 ITC_DEVID_WANRADIO_NONE, 268 ITC_DEVID_WANRADIO_SIEMENS_MC45, 268 ITC_DEVID_WANRADIO_SIERRA_SB555, 268

ITC_DEVID_WANRADIO_XIRCOM_GEM3503, 268 ITC_DHATTR_READFILTER IADC::SetAttribute, rgbData, 157 IBarCodeReaderControl::SetAttribute, 167 ITC_DHDEVFLAG_NODATA, ITCDeviceOpen, 149 ITC_DHDEVFLAG_READAHEAD IADC::Initialize, 153 IBarCodeReaderControl::Initialize, 162 ITCDeviceOpen, 149 ITC_FILE_OPEN_E, IImage::ReadSigCapFile, 224 ITC_IFTP_STOP, 260 ITC_IMGBUFF_TOO_SMALL_E IImage::ReadImage, 225 IImage::ReadSigCapBuffer, 223 ITC INV PARAMETER E IImage::ReadImage, 225 IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 ITC_KEYBOARD_CHANGE, CreateEvent(), 283 ITC_MAXFILTER_CHARS, IBarCodeReaderControl::SetAttribute, 167 ITC_MULTICLIENT_ENABLE, IADC::SetAttribute eAttribID, 157 rgbData, 157 ITC_NVPARM_80211_INSTALLED, 268 ITC_NVPARM_80211_RADIOTYPE, 268 ITC_NVPARM_ANTENNA_DIVERSITY, 267 ITC NVPARM BLUETOOTH INSTALLED, 268 ITC NVPARM CONTRAST, 266 ITC NVPARM DISPLAY TYPE, 266 ITC_NVPARM_ECN, 266 ITC_NVPARM_EDBG_SUBNET, 266 ITC NVPARM EDG IP, 266 ITC_NVPARM_ETHERNET_ID, 265 ITC NVPARM INTERMEC DATACOLLEC-TION_HW, 267 ITC_NVPARM_INTERMEC_DATACOLLEC-**TION_SW**, 267 ITC_NVPARM_INTERMEC_SOFTWARE_CON-**TENT**, 267 ITC_NVPARM_LAN9000_INSTALLED, 269 ITC NVPARM MANF DATE, 265 ITC NVPARM MCODE, 266 ITC_NVPARM_RTC_RESTORE, 267 ITC NVPARM SERIAL NUM, 265 ITC_NVPARM_SERIAL2_INSTALLED, 269 ITC_NVPARM_SERVICE_DATE, 266 ITC NVPARM SIM PROTECT HW INSTALLED, ITC NVPARM SIM PROTECT SW INSTALLED, 2.69 ITC_NVPARM_VERSION_NUMBER, 266 ITC_NVPARM_VIBRATE_INSTALLED, 269 ITC_NVPARM_WAN_FREQUENCY, 268 ITC NVPARM WAN INSTALLED, 268

ITC NVPARM WAN RADIOTYPE, 268

ITC_NVPARM_WAN_RI, 267 ITC_RDRATTR_GOOD_READ_BEEP_DURATION, IBarCodeReaderControl::SetAttribute, 167 ITC RDRATTR GOOD READ BEEPS NUMBER, IBarCodeReaderControl::SetAttribute, 167 ITC_RDRATTR_GOOD_READ_LED_ENABLE, IBar-CodeReaderControl::SetAttribute, 167 ITC_RDRATTR_SCANNER_ENABLE, IBarCodeReaderControl::SetAttribute, 167 ITC RDRATTR TONE ENABLE, IBarCodeReader-Control::SetAttribute, 167 ITC_RDRATTR_TONE_FREQUENCY, IBarCodeReaderControl::SetAttribute, 167 ITC_RDRATTR_VOLUME_LEVEL, IBarCodeReader-Control::SetAttribute, 167 ITC REGISTRY LOCATION, 270 ITC REGISTRY SAVE ENABLE, 270 ITC_RESULT_ERR_BADREGION E IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 ITC_RESULT_NO_BC_DECODED_E IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 ITC_TIMEOUT_E, IImage::ReadImage, 225 ITU-T interface specifications, 115

Κ

Keeping a to-do list, via Tasks, 37 KernelIoControl IOCTL_GET_CPU_ID, 280 IOCTL_HAL_COLDBOOT, 275, 280 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274 IOCTL_HAL_GET_DEVICE_INFO, 264 IOCTL_HAL_GET_DEVICEID, 272 IOCTL HAL GET OAL VERINFO, 273 IOCTL HAL GET RESET INFO, 276 IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 IOCTL HAL REBOOT, 278, 280 IOCTL_HAL_WARMBOOT, 275, 280 **IOCTL PROCESSOR INFORMATION, 279** Key clicks bar code configuration, 370 configuration parameter, 354 Keyboard, Pocket PC input panel, 17 Keypad advanced remapping, 283 change notification, 283 driver registry settings, 282 planes, 281 remapping, 281 sample registry keys, 284 Knowledge Central, xviii

L

Laser scanner configuration parameters, 286 data collection parameters beeper frequency, 320 beeper volume, 318 codabar, 292 codabar user ID, 310 code 11, 306 code 11 user ID, 314 code 128, 295 code 128 FNC1 character, 297 code 128 user ID, 310 code 39, 290 code 39 user ID, 310 code 93, 294 code 93 length, 294 code 93 user ID, 311 EAN-13 user ID, 313 EAN-8 user ID, 313 good read beep duration, 322 good read beeps, 321 interleaved 2 of 5, 303 interleaved 2 of 5 user ID, 311 macro PDF, 300 matrix 2 of 5, 304 matrix 2 of 5 user ID, 313 micro PDF 417, 302 MSI, 299 MSI user ID, 311 PDF 417, 300 PDF 417 user ID, 311 plessey, 298 plessey user ID, 312 prefix, 315 standard 2 of 5, 291 standard 2 of 5 user ID, 312 suffix, 316 telepen, 305 telepen user ID, 313 UPC-E user ID, 312 UPC-A user ID, 312 **UPC/EAN**, 293 virtual wedge, 325 virtual wedge code page, 329 virtual wedge grid, 328 virtual wedge postamble, 327 virtual wedge preamble, 326 SNMP configuration parameters identification contact, 338 identification location, 340 identification name, 339 security encryption key, 335 security read encryption, 333 security read-only community string, 331 security read/write community string, 332

security subnet mask, 309 security write encryption, 334 trap authentication, 336 trap threshold, 337 supported beeper functions, 317 symbologies, 289 supported symbologies, 147 symbologies not available data matrix, 308 Datamatrix, 307 LEAP, 802.1x profile, security information, 93 Letter recognizer, Pocket PC input panel, 18 Library, Microsoft Reader, 59 Line printing, 128 **lpBytesReturned** IOCTL_GET_CPU_ID, 280 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274IOCTL_HAL_GET_DEVICE_INFO, 264 IOCTL_HAL_GET_DEVICEID, 272 IOCTL_HAL_GET_OAL_VERINFO, 273 IOCTL_HAL_GET_RESET_INFO, 276 IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 IOCTL_PROCESSOR_INFORMATION, 279 lpInBuf IOCTL GET CPU ID, 280 IOCTL HAL COLDBOOT, 275 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274 IOCTL_HAL_GET_DEVICE_INFO, 264 IOCTL_HAL_GET_DEVICEID, 272 IOCTL_HAL_GET_OAL_VERINFO, 273 IOCTL_HAL_GET_RESET_INFO, 276 IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 IOCTL_HAL_REBOOT, 278 IOCTL_HAL_WARMBOOT, 275 IOCTL_PROCESSOR_INFORMATION, 279 lpInBufSize IOCTL_GET_CPU_ID, 280 IOCTL HAL COLDBOOT, 275 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274 IOCTL_HAL_GET_DEVICE_INFO, 264 IOCTL_HAL_GET_DEVICEID, 272 IOCTL_HAL_GET_OAL_VERINFO, 273 IOCTL_HAL_GET_RESET_INFO, 276 IOCTL_HAL_REBOOT, 278 IOCTL_HAL_WARMBOOT, 275 **IOCTL PROCESSOR INFORMATION**, 279

lpOutBuf IOCTL_GET_CPU_ID, 280 IOCTL_HAL_COLDBOOT, 275 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274IOCTL_HAL_GET_DEVICE_INFO, 264 IOCTL_HAL_GET_DEVICEID, 272 IOCTL_HAL_GET_OAL_VERINFO, 273 IOCTL_HAL_GET_RESET_INFO, 276 IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 IOCTL_HAL_REBOOT, 278 IOCTL_HAL_WARMBOOT, 275 **IOCTL PROCESSOR INFORMATION, 279** LPT9 printer device, 129

Μ

Macro PDF, configuration parameter, 300 Managing e-mail messages and folders, via Inbox, 46 ManifestName, FTP Server, 254 Matrix 2 of 5 configuration parameter, 304 user ID, 313 default S9C settings, 186 enumerations, 186 IS9CConfig::GetMatrix2of5, 185 IS9CConfig::SetMatrix2of5, 186 modifier characters, 219 Meetings, via Calendar, 31 Menus, Pocket PC settings, 26 MIBs ASN.1, 125 files, 125 object identifier, 126 OIDs, 126 Micro PDF 417, configuration parameter, 302 Microphone, internal, 2 Microsoft Developer Network Library. See MSDN library Microsoft Exchange e-mail account, 53 Microsoft Passport account, 53 Microsoft Reader books downloading, 58 reading, 60 removing, 61 features, 61 adding bookmarks, 61 adding drawings, 61 annotations index, 61 attaching notes, 61 copying text, 61 highlighting text, 61 searching for text, 61 Pocket PC, 58 using the library, 59 Microsoft Word. See Pocket Word Migrating applications, 80 Mini-Landline modems, 4 Mobile Favorites folder, Pocket Internet Explorer, 62

Modems card support, 4 creating a connection to an ISP, 68 to work, 70 MP3 files, Windows Media Player, 57 MSDN library, 260 MSI, 367 configuration parameter, 299 user ID, 311 default S9C settings, 187 enumerations, 188 IS9CConfig::GetMSI, 187 IS9CConfig::SetMSI, 187 modifier characters, 219 MSN account, 53 MSN Messenger about, 53 accounts Hotmail, 53 Microsoft Exchange e-mail, 53 Microsoft Passport, 53 MSN, 53 contacts chatting with, 55 working with, 54 Pocket PC icon, 13 setting up, 54 using My Text, 24 MultiMediaCards, card support, 5

Ν

nBufferSize, IADC::SetAttribute, 157 nDepth, IImage::ReadImage, 225 NDIS_NET_AUTO_UNKNOWN, GetNetworkMode(), 103 NDIS_NET_MODE_ESS, GetNetworkMode(), 103 NDIS NET MODE IBSS, GetNetworkMode(), 103 NDIS_NET_MODE_UNKNOWN, GetNetworkMode(), 103 NDIS_NET_TYPE_DS, GetNetworkType(), 102 NDIS_NET_TYPE_FH, GetNetworkType(), 102 NDIS_NET_TYPE_UNDEFINED, GetNetworkType(), 102 NDIS_POWER_LEVEL_1, GetTXPower(), 102 NDIS_POWER_LEVEL_15, GetTXPower(), 102 NDIS_POWER_LEVEL_30, GetTXPower(), 102 NDIS_POWER_LEVEL_5, GetTXPower(), 102 NDIS_POWER_LEVEL_63, GetTXPower(), 102 NDIS_POWER_LEVEL_UNKNOWN, GetTXPower(), 102NDIS_RADIO_ASSOCIATED, GetAssocationStatus(), 104NDIS_RADIO_AUTH_MODE_AUTO, GetAuthenticationMode(), 104 NDIS_RADIO_AUTH_MODE_ERROR, GetAuthenticationMode(), 104 NDIS_RADIO_AUTH_MODE_OPEN, GetAuthenticationMode(), 104

Index

NDIS_RADIO_AUTH_MODE_SHARED, GetAuthenticationMode(), 104 NDIS_RADIO_POWER_MODE_CAM, GetPower-Mode(), 105 NDIS_RADIO_POWER_MODE_MAX, GetPower-Mode(), 105 NDIS_RADIO_POWER_MODE_PSP, GetPowerMode(), 105NDIS RADIO POWER UNKNOWN, GetPower-Mode(), 105 NDIS_RADIO_SCANNING, GetAssociationStatus(), 104 NDIS_RADIO_WEP_ABSENT, GetWepStatus, 104 NDIS_RADIO_WEP_DISABLED, GetWepStatus(), 104 NDIS_RADIO_WEP_ENABLED, GetWepStatus(), 104 NDIS RADIO WEP NOT SUPPORTED, GetWepStatus(), 104 Network adapters 802.11b, 87 antenna color code, 85 Ethernet communications, 86 wireless printing, 120 WWAN radio options, 110 Network management. See SNMP Network type, 802.11 radio module, 89 nFilterChars IADC::SetAttribute, 157 IBarCodeReaderControl::SetAttribute, 167 nInBufSize IOCTL HAL ITC READ PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 Notes creating a note, 41 drawing on the screen, 22 creating a drawing, 22 selecting a drawing, 22 Pocket Outlook, 40 Pocket PC icon, 13 recording a message, 23 creating a recording, 23 writing on the screen, 19 alternate writing, 21 converting writing to text, 20 selecting the writing, 19 tips for good recognition, 21 nOutBufSize IOCTL_GET_CPU_ID, 280 IOCTL_HAL_COLDBOOT, 275 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274 IOCTL_HAL_GET_DEVICE_INFO, 264 IOCTL_HAL_GET_DEVICEID, 272 IOCTL_HAL_GET_OAL_VERINFO, 273 IOCTL_HAL_GET_RESET_INFO, 276 IOCTL HAL ITC READ PARM, 265 IOCTL HAL ITC WRITE SYSPARM, 270 IOCTL_HAL_REBOOT, 278

IOCTL_HAL_WARMBOOT, 275 IOCTL_PROCESSOR_INFORMATION, 279 NPCP printing, 129 about, 129 closing driver, 130 COM1 parameters, 129 communications, 132 driver I/O controls, 131 installation, 129 LPT9, 129 opening driver, 130 reading from driver, 130 removal, 129 sample code, 132 unit information control panel, NPCPTEST CAB file, 344 writing to driver, 130 0 O'Neil printing See also DTR printer installing driver, 134 Object store IOCTL_HAL_COLDBOOT, 275 IOCTL_HAL_REBOOT, 278 IOCTL_HAL_WARMBOOT, 275 Oldstyle device ID, 272 Onsite repair, xviii Opening drivers DTR, 135 NPCP, 130 Operators, virtual wedge grid, 144 Other publications, xviii Owner information, Pocket PC settings, 26 Ρ Page format printing, 128 Password Pocket Excel, 53 Pocket PC settings, 26 PauseAtStartup, FTP Server, 255 pBarCodeDataDetails, IBarCodeReaderControl::Read, 165 pbyFNC1, IS9CConfig::GetCode128, 180 PDF 417 about the laser scanner, 137 configuration parameter, 300 user ID, 311 default S9C settings, 190 enumerations, 190 extensions IS9CConfig2::GetPDF417ext, 213 IS9CConfig2::SetPDF417ext, 213 IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224

pdwBufferSize, IS9CConfig2::GetGlobalAmble, 211

IS9CConfig::GetPDF417, 188

IS9CConfig::SetPDF417, 189

modifier characters, 220

pdwLength IS9CConfig::GetCode128, 180 IS9CConfig::GetCode93, 179 IS9CConfig::GetMatrix2of5, 185 IS9CConfig::GetMSI, 187 IS9CConfig::GetPlessey, 192 pdwNumBytes IS9CConfig::GetCodabar, 173 IS9CConfig::GetI2of5, 183 IS9CConfig::GetStandard2of5, 194 pdwNumElement, IS9CConfig2::GetCustomSymIds, 207 pdwReplyBuffSize ISCP::GetConfig, 217 ISCP::SetConfig, 218 pdwTotalDiscardedBytes, IADC::CancelReadRequest, 152 peCheck IS9CConfig::GetCodabar, 173 IS9CConfig::GetCode39, 176 IS9CConfig::GetI2of5, 183 IS9CConfig::GetMSI, 187 IS9CConfig::GetPlessey, 192 IS9CConfig::GetStandard2of5, 194 IS9CConfig2::GetCode11, 205 peCip128State, IS9CConfig::GetCode128, 180 peCLSI, IS9CConfig::GetCodabar, 173 peCode128, IS9CConfig2::GetPDF417Ext, 213 peDecode IS9CConfig::GetCodabar, 173 IS9CConfig::GetCode128, 180 IS9CConfig::GetCode39, 176 IS9CConfig::GetCode93, 179 IS9CConfig::GetI2of5, 183 IS9CConfig::GetMatrix2of5, 185 IS9CConfig::GetMSI, 187 IS9CConfig::GetPlessey, 192 IS9CConfig::GetStandard2of5, 194 IS9CConfig::GetTelepen, 197 IS9CConfig2::GetCode11, 205 IS9CConfig2::GetPDF417Ext, 213 peEan128Ident, IS9CConfig::GetCode128, 180 peFormat IS9CConfig::GetCode39, 176 IS9CConfig::GetStandard2of5, 194 IS9CConfig::GetTelepen, 197 peLengthId IS9CConfig::GetCodabar, 173 IS9CConfig::GetI2of5, 183 IS9CConfig::GetStandard2of5, 194 peMacroPdf, IS9CConfig::GetPDF417, 188

pePdf417Decode, IS9CConfig::GetPDF417, 188 pePdfAddressee, IS9CConfig::GetPDF417, 189 pePdfChecksum, IS9CConfig::GetPDF417, 189 pePdfControlHeader, IS9CConfig::GetPDF417, 188 pePdfFileName, IS9CConfig::GetPDF417, 188 pePdfFileSize, IS9CConfig::GetPDF417, 189 pePdfSegmentCount, IS9CConfig::GetPDF417, 188 pePdfSender, IS9CConfig::GetPDF417, 189 pePdfTimeStamp, IS9CConfig::GetPDF417, 188 peSS IS9CConfig::GetCodabar, 173 IS9CConfig::GetCode39, 176 peSSChars, IS9CConfig::GetCode39, 176 peSymIdXmit, IS9CConfig2::GetSymIdXmit, 214 peVer, IS9CConfig2::GetCode11, 205 pImgBuffer IImage::ReadImage, 225 IImage::ReadSigCapBuffer, 222 Planes, keypad, 281 Plessey, 367 configuration parameter, 298 user ID, 312 default S9C settings, 193 enumerations, 193 IS9CConfig::GetPlessey, 192 IS9CConfig::SetPlessey, 192 modifier characters, 220 pnBufferData, IADC::QueryAttribute, 154 pnBytesReturned IADC::Read, 156 IBarCodeReaderControl::Read, 165 Pocket Excel about, 52 creating a workbook, 52 Pocket PC icon, 13 Pocket Internet Explorer about, 62 adding programs, 27 AvantGo channels, 64 browing the Internet, 66 favorite links, 62 getting connected, 67 Mobile Favorites folder, 62 Pocket PC icon, 13 software build, 5 viewing mobile favorites and channels, 66 Pocket Outlook, 31 Calendar, 31

Pocket PC about, 8 ActiveSvnc, 29 basic skills, 11 Calendar, 31 command bar, 14 Contacts, 34 edition information, 8 getting connected, 67 Inbox, 42 input panel. See Input panel IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274 IOCTL_HAL_GET_OAL_VERINFO, 273 MSN Messenger, 53 navigation bar, 14 Notes, 40 notifications, 15 status icon, 12 Pocket Excel, 52 Pocket Word, 47 pop-up menus, 15 programs, 13 status icons, 12 support URLs, 10 Tasks, 37 Today screen, 11 where to find information, 10 Windows Media Player, 57 writing on the screen, 19 Pocket Word about, 47 creating a document, 47 drawing mode, 51 Pocket PC icon, 13 recording mode, 51 tips, 53 typing mode, 49 writing mode, 50 POP3, Folder behavior connected to e-mail server, 46 Postamble configuration parameter, 327 with/without data, 327 Power control panel, battery status, 2 Pocket PC settings, 26 ppvObject ITCDeviceClose, 150 ITCDeviceOpen, 149 Preamble configuration parameter, 326 with/without data, 326 Prefix, configuration parameter, user ID, 315 Printer support, 128 IrDA printer driver, 128 NPCP printer driver, 129

O'Neil printer driver, 134 Processor information, IOCTL_PROCESSOR_IN-FORMATION, 279 Profiles 802.11 radio module, 87 basic information, 89 certificates, 95 import/export, 96 read-only, 94 scan list, 97 security information, 90 adding to unit, 88 deleting, 88 editing, 88 Programs, adding or removing, Pocket PC, 26 pSigCapSpec IImage::ReadSigCapBuffer, 221 IImage::ReadSigCapFile, 224 pStructSymIdPair IS9CConfig2::GetCustomSymIds, 207 IS9CConfig2::SetCustomSymIds, 208 pSystemTime, IADC::Read, 156 pszDevice, ITCDeviceOpen, 149 pszDeviceName IADC::Initialize, 153 IBarCodeReaderControl::Initialize, 162 pszFileName, IImage::ReadSigCapFile, 224 pwLength, IS9CConfig::GetCode39, 176 pwTotalDiscardedBytes, IBarCodeReaderControl::Cancel-ReadRequest, 160 pwTotalDiscardedMessages IADC::CancelReadRequest, 152 IBarCodeReaderControl::CancelReadRequest, 160

QR code configuration parameter, 307 IS9CConfig3 function, 216 Quick Response code. See QR code

R

Radios See also Network adapters card support, 4 Reader commands, 357 configuration change, 357 date and time settings, 358 Reading from drivers, NPCP, 130 Reboot methods IOCTL_HAL_COLDBOOT, 280 IOCTL_HAL_REBOOT, 280 IOCTL_HAL_WARMBOOT, 280 Record button, recording a message, 23 Recording a message See also Notes Pocket Word, 51 Recording mode, Pocket Word, 51

Recovery CD AutoCab method, 82 AutoRun system, 76 AUTOUSER.DAT file, 81 part numbers, xviii RegFlushKey() API, 262 S9C upgrade, 345 updating the system software, 79 RegFlush utility, 81 Registry FTP Server parameters, 252 keypad remapping, 282 sample view of key mapping, 284 save location, IOCTL_HAL_ITC_WRITE_SYSPARM, 270writing to a storage card, 81 Registry settings AutoCfg, 122 AutoFTP, 261 AutoInterval, 122 AutoIP/DHCP, 122 DhcpMaxRetry, 122 DhcpRetryDialogue, 122 EnableDHCP, 122 keypad driver, 282 keypad planes alpha, 282 gold, 282 unshifted, 282 Related publications, xviii Removeable card support, 5 Removing drivers DTR, 134 NPCP, 129 Removing programs, Pocket PC, 26, 28 RFC 959, 259 rgbAttrBuffer IBarCodeReaderControl::QueryAttribute, 164 IBarCodeReaderControl::SetAttribute, 168 rgbBuffer IADC::QueryAttribute, 154 IS9CConfig2::GetGlobalAmble, 211 IS9CConfig2::SetGlobalAmble, 212 rgbCommandBuff ISCP::GetConfig, 217 ISCP::SetConfig, 218 rgbData, IADC::SetAttribute, 157 rgbDataBuffer IADC::Read, 156 IBarCodeReaderControl::Read, 165 rgBeepRequests, IBarCodeReaderControl::IssueBeep, 163 rgbImageData, pImgBuffer, IImage::ReadSigCapBuffer, 222 rgbLengthBuff IS9CConfig::GetCodabar, 173 IS9CConfig::GetI2of5, 183 IS9CConfig::GetStandard2of5, 194 IS9CConfig::SetCodabar, 174

IS9CConfig::SetI2of5, 184 IS9CConfig::SetStandard2of5, 195 rgbReplyBuff ISCP::GetConfig, 217 ISCP::SetConfig, 218 Root, FTP Server, 255

S

S 2 of 5. See Standard 2 of 5 S_DEVICE_CONTENTION_E, IImage::Open, 227 S_DEVICE_NOT_OPENED_E IImage::CancelReadImage, 226 IImage::Close, 228 IImage::ReadImage, 225 IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 IImage::Start, 226 IImage::Stop, 227 S_IMG_NOT_PRESENT_E IImage::Open, 227 IImage::Stop, 227 S OK IImage::CancelReadImage, 226 IImage::Close, 228 IImage::Open, 227 IImage::ReadImage, 225 IImage::ReadSigCapBuffer, 223 IImage::ReadSigCapFile, 224 IImage::Start, 226 IImage::Stop, 227 S9C initialization, 138 IS9CConfig functions, 172 IS9CConfig2 functions, 204 IS9CConfig3 functions, 216 unit information control panel, upgrade files, 345 version number, 342 Sabre 1551E or 1553 See also Tethered scanner cabling, 232 settings, 232 Sample code, NPCP printing, 132 SB555 radio, 110 Scan list of profiles, 802.11 radio module, 98 Scanner, unit configuration parameters automatic shutoff, 355 backlight timeout, 353 date/time, 352 key clicks, 354 volume, 356 Scanner cabling, 232 Scheduling appointments and meetings, via Calendar, 31 SDK, unit information control panel, 345 SDMMC Disk, 80 Searching for text, Microsoft Reader, 61 SecureDigital cards card support, 5 installing applications, 78, 79

Security configuration parameter encryption key, 335 read encryption, 333 read-only community string, 331 read/write community string, 332 subnet mask, 309 write encryption, 334 Unit Manager, 350 Selected profile, 802.11 radio module, 97 Selecting, drawing via Notes, 22 Sending and receiving messages, via Inbox, 42 Serial port, modem support, 4 Service contract status, xviii Setting date and time, 358 Setting up an e-mail service, 73 SETUP.DLL, installation functions, 248 Signature capture IImage interface, 221 IImage::CancelReadImage, 226 IImage::Close, 228 IImage::Open, 227 IImage::ReadImage, 225 IImage::ReadSigCapBuffer, 221 IImage::ReadSigCapFile, 224 IImage::Start, 226 IImage::Stop, 227 Simple Network Management Protocol. See SNMP Site installations, xviii Site surveys, xviii SMS, Folder behavior connected to e-mail server, 46 **SNMP**, 125 about SNMP, 123 CMIP, 123 configuration parameters identification contact, 338 identification location, 340 identification name, 339 security encryption key, 335 security read encryption, 333 security read-only community string, 331 security read/write community string, 332 security subnet mask, 309 security write encryption, 334 trap authentication, 336 trap threshold, 337 control primitives, 123 multiple retrievals, 124 retrieval, 124 Unit Manager, 350 using the protocol, 123 SNMP OIDs aimer LED duration, 323 automatic shutoff, 355 backlight timeout, 353 beeper frequency, 320 volume, 318 codabar, 292 user ID, 310

code 11, 306 user ID, 314 code 128, 295 FNC1 character, 297 user ID, 310 code 39, 290 user ID, 310 code 93, 294 length, 294 user ID, 311 data matrix, 308 date/time, 352 default S9C settings, 210 EAN 13 user ID, 313 8 user ID, 313 good read beep duration, 322 beeps, 321 identification contact, 338 location, 340 name, 339 image dimension, 324 interleaved 2 of 5, 303 user ID, 311 IS9CConfig2::GetCustomSymIds, 207 IS9CConfig2::GetSymIdXmit, 214 IS9CConfig2::SetCustomSymIds, 208 IS9CConfig2::SetSymIdXmit, 214 key clicks, 354 macro PDF, 300 matrix 2 of 5, 304 user ID, 313 micro PDF 417, 302 MSI, 299 user ID, 311 PDF 417, 300 user ID, 311 plessey, 298 user ID, 312 prefix, 315 QR code, 307 security encryption key, 335 read encryption, 333 read-only community string, 331 read/write community string, 332 write encryption, 334 security subnet mask, 309 standard 2 of 5, 291 user ID, 312 suffix, 316 telepen, 305 user ID, 313 transmission option, 215 trap authentication, 336 threshold, 337

UPC A user ID, 312 E user ID, 312 **UPC/EAN**, 293 virtual wedge, 325 code page, 329 grid, 328 postamble, 327 preamble, 326 volume, 356 Symbology ID defaults, 219 Software Developer's Kit. See SDK Software Tools CD. See Tools CD Software versions, 6, 342 700 Series Computer, 5 unit information control panel applet, 342, 344 Speaker, 2 SSID (network name), 802.11 radio module, 89 Standard 2 of 5, 366 configuration parameter, 291 user ID, 312 default S9C settings, 196 enumerations, 196 IS9CConfig::GetStandard2of5, 194 IS9CConfig::SetStandard2of5, 195 modifier characters, 220 Start Menu, adding programs, 28 via ActiveSync, 28 via File Explorer, 28 Status icons, Pocket PC, 12 Stream device driver NPCPPORT.DLL, 129 ONEIL.DLL, 134 stTimeStamp, IBarCodeReaderControl::Read, 165 Suffix, configuration parameter, 316 Summary screen Calendar, 33 Contacts, 36 Tasks, 39 Support global services and support center, xviii web, xviii Symbologies, 360 scanning labels, 368 Unit Manager, 348 user IDs Codabar, 310 Code 11, 314 Code 128, 310 Code 39, 310 Code 93, 311 EAN 13, 313 EAN 8, 313 Interleaved 2 of 5, 311 Matrix 2 of 5, 313 MSI, 311 PDF 417, 311 Plessey, 312 Standard 2 of 5, 312 Telepen, 313

UPC A, 312 UPC E, 312 when not available imager, 298, 299, 300, 302, 304, 305, 306 laser scanner, 307, 308 Symbology ID, Unit Manager, 349 Synchronizing e-mail messages, via Inbox, 42 System status maintained, 4 szDest, EncryptWepKeyForRegistry(), 106 szFilter IADC::SetAttribute, 157 IBarCodeReaderControl::SetAttribute, 167 szSource, EncrypWepKeyForRegistry(), 106

T

Tasks creating a task, 38 Pocket Outlook, 37 Pocket PC icon, 13 using the summary screen, 39 TCP/IP client, DHCP server, 122 Technical support, xviii Telepen configuration parameter, 305 user ID, 313 default S9C settings, 197 enumerations, 198 IS9CConfig::GetTelepen, 197 IS9CConfig::SetTelepen, 197 modifier characters, 220 Telephone numbers, xviii Testing AT commands, 117 Tethered scanner capabilities, 233 disabling, 231 enabling, 231 error messages, 232 limitations, 233 settings, 231 Text messages, Pocket PC, 24 Time, setting, 358 Tips for working, Pocket Excel, 53 TLS, 802.1x profile, 92 Today, Pocket PC settings, 26 Today screen, Pocket PC, 11 Tools CD base operating system files, 78 Bluetooth documentation, 120 CAB files, 4, 78, 344 CE Imager, 80 Comm Port Wedge CAB file, 344 management tools installed on desktop, 77 MIB files, 125 part number, xviii sample NPCP code, 132 SRDEVMGMT.CAB file, 286 Unit Manager CAB file, 347 Wireless Printing Development Guide, 120 wireless printing sample, 345 wireless printing SDK, 120

Tracking people, via Contacts, 34 Transcriber, Pocket PC input panel, 18 Transfer items using infrared getting connected, 67 receiving information, 67 sending information, 67 Trap configuration parameters authentication, 336 threshold, 337 Traps control panel appet, SNMP, 336 Unit Manager, 351 Troubleshooting, CAB Wizard, 250 TTLS, 802.1x profile, 93 Typing mode, Pocket Word, 49 Typing on the screen, Pocket Word, 49

U UDP

FTPDCE, 256 within SNMP, 123 UDP broadcasts, IDNATarget parameter, 254 uiHeight, pSigCapSpec, IImage::ReadSigCapBuffer, 222 uiWidth, pSigCapSpec, IImage::ReadSigCapBuffer, 222 Unit, configuration parameters automatic shutoff, 355 backlight timeout, 353 date/time, 352 key clicks, 354 volume, 356 Unit information battery status, 343 CAB files, 344 ActiveX control tools, 345 Bluetooth stack, 344 Comm Port Wedge, 344 NPCP printer, 344 S9C Upgrade, 345 SDK, 345 Unit Manager, 345 Unit Manager help, 345 Windows configuration, 345 wireless printing demo, 345 wireless printing sample, 345 versions, 6, 342 700 Platform Build, 342 DataCollection Build, 342 S9C, 342 Unit Manager automatic shutoff, 355 backlight timeout, 353 data collection, 348 beeper/LED, 349 imager, 350 symbologies, 348 symbology ID, 349 virtual wedge, 350 date/time, 352 documentation, 347 key clicks, 354

reader commands, 357 changing configuration, 357 setting time and date, 358 SNMP, 350 identification, 351 security, 350 traps, 351 volume, 356 Unit manager installing applications, 78 unit information control panel, 345 help files, 345 Universal Product Code. See UPC Unshifted plane on keypad, 281 UPC, 362 configuration parameter, 293 A user ID, 312 E user ID, 312 default S9C settings, 201 enumerations, 201 IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 modifier characters, 220 upcACheck IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 upcAddOn2 IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 upcAddOn5 IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 upcAddOnDigits IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 upcANumSystem IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 upcAReencode IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 upcASelect IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 upceanDecode IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 upcECheck IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 upcENumSystem IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 upcEReencode IS9CConfig::GetUpcEan, 199 IS9CConfig::SetUpcEan, 200 upcESelect IS9CConfig::GetUpcEan, 198 IS9CConfig::SetUpcEan, 200 Updating, bootloader, 77
URLs ActiveSync, 29 Adobe Acrobat Reader, 116 AT command interface CDMA/1xRTT SB555, 116 GPRS/GSM GEM350X, 116 GPRS/GSM MC45, 116 customer support, xviii full screen display, 263 Knowledge Central, xviii MIBs, 125 Microsoft Exchange e-mail account, 53 Microsoft Passport account, 53 Microsoft support, 10 MSDN library, 260 Pocket PC, 10 Pocket PC support, 10 web support, xviii User Datagram Protocol. See UDP Using a message list, via Inbox, 43 UUID, 272

V

Video files, Windows Media Player, 57 Viewing mobile favorites and channels, Pocket Internet Explorer, 66 Virtual wedge bar code configuration grid, 371 postamble, 371 preamble, 371 configuration parameter, 325 code page, 329 grid, 328 postamble, 327 preamble, 326 data filtering, 141 filter expression values, 142 global amble IS9CConfig2::GetGlobalAmble, 211 IS9CConfig2::SetGlobalAmble, 212 operators, 144 Unit Manager, 350 VN_CLASS_ASIC, 266 VN_CLASS_BOOTSTRAP, 266 VN_CLASS_KBD, 266 Volume bar code configuration, 368 configuration parameter, 356 W WAN radio CORE module, 111 WAN radio IDs ITC_DEVID_WANRADIO_NONE, 268 ITC_DEVID_WANRADIO_SIEMENS_MC45, 268 ITC_DEVID_WANRADIO_SIERRA_SB555, 268

ITC_DEVID_WANRADIO_SIE ITC_DEVID_WANRADIO_SIE WAN rado CORE module phone application, 115

terminal application, 114 WAP pages, 62 connecting to an ISP, 68 Warm boot IOCTL_HAL_REBOOT, 278 IOCTL_HAL_WARMBOOT, 275 Web addresses. See URLs Web browsers, FTP support, 259 Web pages, 62 connecting to an ISP, 68 Web support, xviii Welch Allyn 1470 Imager cabling, 232 settings, 232 WEP encryption, 802.11 radio module, 91 Windows configuration, unit information control panel, WinCfg CAB file, 345 Windows Media files, Windows Media Player, 57 Windows Media Player, Pocket PC, 57 Wireless Application Protocol. See WAP pages Wireless network, 87 Wireless printing Bluetooth compatible module, 120 unit information control panel PDWPM0C CAB file, 345 WP_SAMPLE.CAB file, 345 Wireless TCP/IP installations, BlockSize parameter, 252 Wireless WAN AT command interface CDMA/1xRTT SB555, 116 GPRS/GSM GEM350X, 116 GPRS/GSM MC45, 116 CDMA/1xRTT, 110 GEM350x, 110 GSM/GPRS, 110 SB555, 110 testing AT commands, 117 wNumberOfMessages, IADC::QueryData, 155 Word documents. See Pocket Word Work creating a modem connection, 70 an Ethernet connection, 71 getting connected, 70 Writing mode, Pocket Word, 50 Writing on the screen See also Notes Pocket Word, 50 Writing to drivers DTR, 135 NPCP, 130 wStructSize, IBarCodeReaderControl::Read, 165 WWAN. See Wireless WAN

Х

Xscale processor ID, IOCTL_GET_CPU_ID, 280

Files Index

Numbers

80211API.DLL, 100 80211CONF.EXE, 100 80211SCAN.EXE, 100

A

AUTOUSER.DAT, 79

С

CABWIZ.DDF, 249 CABWIZ.EXE, 236, 249 CEIMAGER.EXE, 80 CPL802.CPL, 100

D

DEVICEID.H, 272

Е

EXITME.BIN, 259

F

FTPDCE.EXE, 256, 259 AutoFTP, 261 FTP Server, 251 FTPDCE.TXT, 259

I

IADC.H, IADC functions, 151 IADCDEVICE.H IADC::SetAttribute, 157 IBarCodeReaderControl::SetAttribute, 167 INTERMEC.MIB, 125 IS9CCONFIG.H IS9CConfig functions, 172 IS9CConfig2 functions, 204 ITCADC.MIB, 125 ITCDEVMGMT.H, 149 ITCDEVMGMT.LIB, 149 ITCSNMP.MIB, 125 ITCTERMINAL.MIB, 125 ITCUUID.LIB IADC functions, 151 IS9CConfig functions, 172

IS9CConfig2 functions, 204

М

MAKECAB.EXE, 249 MOD80211.DLL, 100

Ν

NETWLAN.DLL, 100 NPCPPORT.DLL, 129 NRINET.INI, 345

0

OEMIOCTL.H IOCTL_GET_CPU_ID, 280 IOCTL_HAL_COLDBOOT, 275 IOCTL_HAL_GET_BOOT_DEVICE, 277 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 274 IOCTL_HAL_GET_OAL_VERINFO, 273 IOCTL_HAL_GET_RESET_INFO, 276 IOCTL_HAL_ITC_READ_PARM, 265 IOCTL_HAL_ITC_WRITE_SYSPARM, 270 IOCTL_HAL_REBOOT, 278 IOCTL_HAL_REBOOT, 275 ONEIL.DLL, 134

Ρ

PKFUNCS.H IOCTL_HAL_GET_DEVICEID, 272 IOCTL_PROCESSOR_INFORMATION, 279

R

REBOOTME.BIN, 259 __RESETMEPLEASE__.TXT, 248 RPM.EXE, 241 RPMCE212.INI, 241

S

SETUP.DLL, 240, 248 DllMain, 248 SRDEVMGMT.CAB, 286

Т

TAHOMA.TTF, 241

W

WCESTART.INI, 241



 Corporate Headquarters

 6001
 36th Avenue West

 Everett, Washington 98203
 1

 tel
 425.348.2600

 fax
 425.355.9551

 www.intermec.com
 1

700 Series Color Mobile Computer User's Manual - November 2002



961054031 REV B