TEC-IT

WWW.TEC-IT.COM

TWedgeCE

Data Acquisition for Mobile Devices

Version 1.4

User Documentation

August, 14th 2007

TEC-IT Datenverarbeitung GmbH
Wagnerstrasse 6
A-4400 Steyr, Austria

t ++43 (0)7252 72720
f ++43 (0)7252 72720 77
office@tec-it.com
www.tec-it.com

1 Content

1.1 List of Tables

1.2 List of Figures

2 Introduction

2.1 Abstract

TWedgeCE is a generic data acquisition application for Microsoft® Windows® CE based handheld devices. It captures data from an arbitrary device interface and simulates keystrokes based on the captured data. These keystrokes are sent subsequently to an application. So it is possible to input data from external devices - such as bar code readers, electronic scales (and many more) - into Microsoft Pocket Excel®, e.g.

TWedgeCE supports the following interfaces:

- Native RS-232 (serial connection via COM ports)
- Bluetooth® via COM-simulator (usually COM8 and COM9)
- Built-in barcode scanner (only on Symbol® PPT8800)

This document explains the user interface of *TWedgeCE* on PocketPCs and on Microsoft Windows CE devices. Subsequently the program settings are explained.

2.2 Supported Platforms

TWedgeCE is available for Windows CE based operating systems. You can download the following versions on http://www.tec-it.com:

2.2.1 PocketPC

TWedgeCE_PPC2002.exe

This setup can be used for the following operating systems:

- PocketPC 2002
- PocketPC 2003
- Windows Mobile 5.0

2.2.2 Windows CE

TWedgeCE_WinCE42_ARMV4.exe
This version is for ARMV4 processors.

TWedgeCE_WinCE42_ARMV4I.exe
This version is for ARMV4I processors.

TWedgeCE_WinCE42_PPT8800.exe
This version supports the built in barcode-reader of the Symbol PPT 8800.

TWedgeCE_WinCE42_X86.exe
This version is for X86 processors.

These setups can be used for the following operating systems:

- Windows CE 4.x
- Windows CE 5.x

3 User Interface: PocketPC 2002

This chapter describes the user interface on PocketPC systems.

3.1 Main Window

After starting *TWedgeCE* the following screen appears:

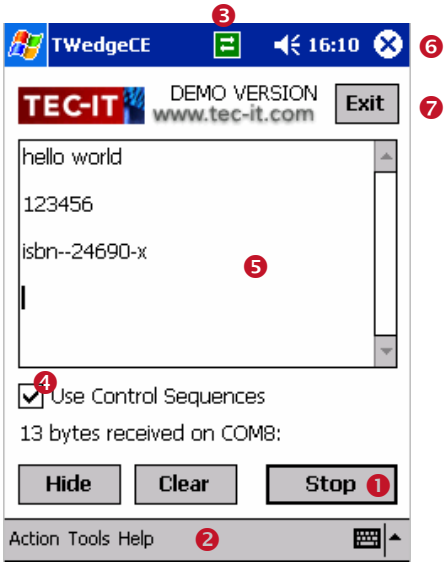

Figure 1: TWedgeCE application window (PocketPC 2002)

The Start/Stop button ❶ allows you to connect/disconnect to the serial device (e.g. to a barcode scanner). If the application is started, the button caption will change to “Stop”. All incoming data is sent to the visible application window.

By default the input will be displayed in the text area of *TWedgeCE* ❺. If you want the input passed to another application (e.g. Pocket Word) just start the desired application and make sure it is visible on the display of your handheld. The data will be sent to the actual position of the text cursor. (Prefix, postfix, delimiter settings and the translation table are applied as described under 0, 5.2.4 and 5.2.5).

Checkbox ❹ shows, whether control sequences are enabled or not (these sequences can be adjusted in the ini-file). If you change the check box, please consider that the state of the control will NOT be saved for the next start-up (change the settings-file to edit start up options). For more information on how control sequences will affect the output see 5.2.5.

The menu ❷ allows you to start/stop and hide *TWedgeCE* (submenu “Action”). Under “Tools” you can edit the program settings or bring up the debug window. Choosing “Help” you will find some information about the product.

The Tray Icon ❸ shows the status of the program. The green icon signals *TWedgeCE* is waiting for data. The red icon indicates *TWedgeCE* is not connected (stopped). By clicking on the icon you can bring down a popup-menu. This menu allows you to start, stop, show, hide or exit *TWedgeCE*. The icon can be turned off on demand. (see 5.2.7, Settings for the Tray Menu).

By clicking (X) ❻ you can hide the application. (Same as selecting hide from menu ❷ or from the tray icon-menu ❸). Finally, ❼ closes *TWedgeCE*.

3.2 Debug Window

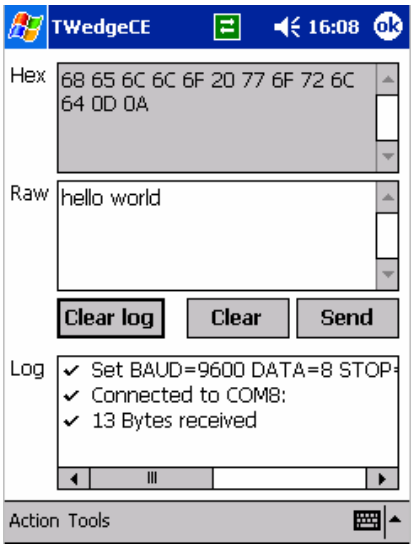

Figure 2: TWedgeCE Debug Window (PocketPC 2002)

TWedgeCE allows you to view some additional information about the received data on the debug screen. You can bring up this screen by clicking Tools → Debug… in menu ❷.

- The Hex box displays the input in hexadecimal codes.
- The Raw box displays the received data an plain text.
- The Log window monitors some information about the connection.
- The Send button sends the content of the Raw box to the open connection.
  Note: For sending data the connection must support writing! (Bluetooth e.g. allows reading on COM8 and sending on COM9 only). Furthermore the external device must be capable of receiving data.

4 User Interface: Windows CE 4.2

This chapter describes the user interface on Windows CE systems.

4.1 Main Window

After starting *TWedgeCE* the following screen appears:

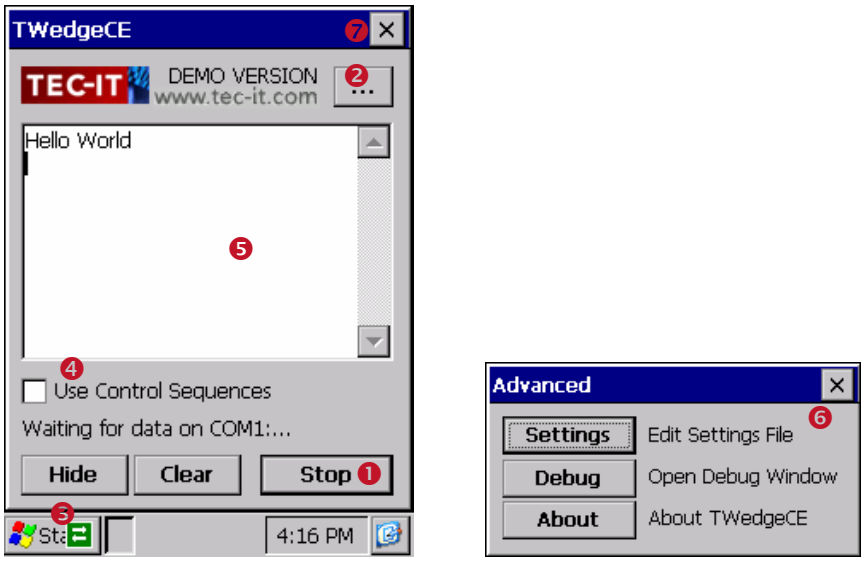

Figure 3: TWedgeCE Application Window (WinCE 4.2)

The Start/Stop button ❶ allows you to connect/disconnect to the serial device (e.g. to a barcode scanner). If the application is started, the button caption will change to “Stop”. All incoming data is sent to the visible application window.

By default the input will be displayed in the text area of *TWedgeCE* ❺. If you want the input passed to another application (e.g. Pocket Word) just start the desired application and make sure it is visible on the display of your handheld. The data will be sent to the actual position of the text cursor. (Prefix, postfix, delimiter settings and the translation table are applied as described under 0, 5.2.4 and 5.2.5).

Checkbox ❹ shows, whether control sequences are enabled or not (these sequences can be adjusted in the ini-file). If you change the check box, please consider that the state of the control will NOT be saved for the next start-up (change the settings-file to edit start up options). For more information on how control sequences will affect the output see 5.2.5.

By clicking ❷ a menu dialog (❻) appears. Using this dialog you can open the program settings, the debug window or the about box.

Icon ❸ shows the status of the program. The green icon signals *TWedgeCE* is waiting for data. The red icon indicates *TWedgeCE* is not connected (stopped). By clicking on the icon you can bring up a popup-menu. This menu allows you to start, stop, show, hide or exit *TWedgeCE*.
The icon can be turned off on demand. (see 5.2.7, Settings for the Tray Menu)

❼ closes *TWedgeCE*.

4.2 Debug Window

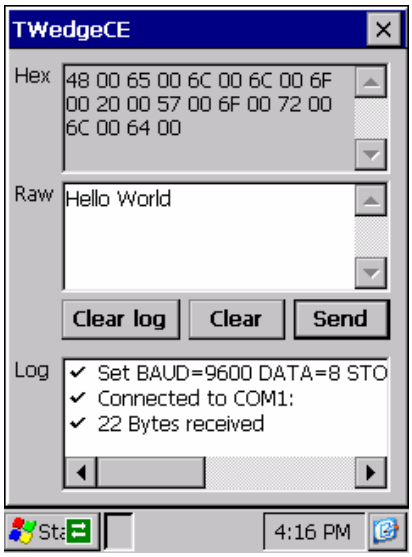

Figure 4: TWedgeCE Debug Window (WinCE 4.2)

TWedgeCE allows you to view some additional information about the received data on the debug screen. You can bring up this screen by clicking … → Debug…

- The Hex box displays the input in hexadecimal codes.
- The Raw box displays the received data as plain text.
- The Log window monitors some information about the connection.
- The Send button sends the content of the Raw box to the open connection.
  Note: For sending data the connection must support writing! (Bluetooth e.g. allows reading on COM8 and sending on COM9 only). Furthermore the external device must be capable of receiving data.

5 Settings

5.1 Introduction

TWedgeCE is customized via a single settings file (.txt extension). This makes it easy to deploy the settings to several mobile devices.

When starting *TWedgeCE* for the first time a standard configuration file will be created. This file contains the following settings:

- Startup options
- Parameters for the serial connection
- Rules for data packaging
- A string replacement-table (TranslationTable)
- Control sequence settings
- Pre-Evaluation expression
- Settings for the tray menu
- Operation mode

► **PocketPC:** You can edit this file by selecting "Tools" → "Settings" from the menu. The file will be opened in the associated text editor (e.g. Pocket Word).

► **Windows CE:** Under Windows CE you have to click ... and then "Settings". A dialog for editing the file appears.

5.2 Settings in Detail

5.2.1 Startup Options

Section [Startup] in the ini-file

AutoStart=N	Automatically connect *TWedgeCE* on startup? When set to "Y" *TWedgeCE* automatically starts to connect as soon as the program is launched. Possible values are: N, Y
AutoHide=N	Hide *TWedgeCE* on startup? When set to "Y", the *TWedgeCE* window will be hidden upon startup. Only the "tray-icon" (see figure 1 or 3, ❸) will be visible – assuming the icon is not turned off (see 5.2.7). Possible values are: N, Y

Table 1: Startup parameters

5.2.2 Parameters for the Serial Connection

Section [Com Settings] in the ini-file

Baud=9600	Baud rate. Common values are: 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600, 115200, 128000, 256000
IOType=3	Serial communication. Reserved for future extension.
Device=COM8:	Device name. Can be either COM1: to COM9: Symbol PPT 8800: Additionally supported is SCN1: (for the built-in hardware scanner of the device)
Data=8	Number of data bits. Possible values are: 7, 8
Stop=1	Number of stop bits. Possible values are: 1, 2
Parity=N	Parity checking. Possible values are: E,O,N
XOnXOff=N	XonXoff handshake on/off. Possible values are: N, Y
RtsCts=N	RtsCts handshake on/off. Possible values are: N, Y
DtrDsr=N	DtrDsr handshake on/off. Possible values are: N, Y
Unicode=N	Receive and transmit unicode strings? (Unicode uses 2 bytes per character.) If Unicode=N ASCII encoding will be used.
ReadTimerInterval=250	Time interval (in milliseconds), in which the communication port is checked for pending data.
RetryCount=1	Retry count. If the connection to the serial device breaks, *TWedgeCE* will try to reconnect for the given number of times before reporting a connection error. This setting can be useful when you lost the connection to a bluetooth device (out of range) or when you turned off your mobile device without closing (and without stopping) *TWedgeCE*. When turning the device back on the connection must be re-established, which may take a couple of retries. Possible values range from -1 to 255. -1 …retry forever 0 …no retry 1..255 …one to 255 retries
RetryTimerInterval=5000	Timer interval for connection retries (in milliseconds). When the connection is lost *TWedgeCE* will wait the given number of milliseconds before trying to reconnect. Possible values range from 1000 to 10000. (1 to 10 seconds).

Table 2: Serial connection parameters

- If the device is capable of Bluetooth communication and if it offers a Bluetooth stack, the Bluetooth communication will be mapped to the serial communication port COM8: for receiving data and to COM9: for sending data by default.
- When using “Device=SCN1:” on the Symbol PPT 8800 all other connection settings will be ignored.

5.2.3 Rules for Data Packaging

Section [Com Settings] / DataPackaging

NoOfBytes=100	Cut input stream after NoOfBytes bytes
Timeout=100	Timeout in milliseconds. All bytes received within Timout ms will be treated as one data package
Timeout_Infinite=N	If this value is set to "Y" the data packaging will be dependant on NoOfBytes and Delimiter only. The Timeout value will be ignored. Possible values are: N, Y
Delimiter=	Delimiter-string
UseDelimiter=N	If enabled, the input stream will be fragmented with every occurrence of the delimiter-string. Possible values are: N, Y
IncludeDelimiter=N	Include the delimiter-string in output stream? Possible values are: N, Y
Prefix=	Prefix (may include control commands)
Postfix=	Postfix (may include control commands)

Table 3: Data packaging parameters

The Rules for data packaging are applied in the following order:

1. Timeout
   All data received within a period of at least Timeout milliseconds is treated as one data package. This data package will be fragmented according to the following rule:
2. BlockSize and Delimiter
   If the data package received within the Timeout period is bigger than NoOfBytes or if a delimiter is found within the first NoOfBytes the data package will be fragmented.
3. The fragments are sent to the application if one of the two rules above apply.

► The Prefix and the Postfix are sent before and after each data fragment.

Example

```
NoOfBytes=10
Timeout_Infinite=N
Delimiter=$
UseDelimiter=Y
IncludeDelimiter=N
```

The scanner reads the following 7 characters: "1234$56".

Since the Delimiter is set to "$" and IncludeDelimiter=N the string "1234" will be passed on to the application. The leftover-string "56" is less than 10 bytes long and does not include the delimiter "$". Thus it is not sent by now.

If some more data is received *within the next Timeout (milliseconds)* – e.g. the scanner sends "789012345678$" - the new data will be appended to the leftover "56" from the last scan. This time the application receives "5678901234" (NoOfBytes=10!). "5678" is stored for another Timeout milliseconds.

Finally, if no more data is received *within the next Timeout,* "5678" will be passed to the application.

Escape Sequences

You can use escape sequences in the Prefix, in the Postfix, in the TranslationTable (left-hand value and right-hand value) and you can use escape sequences in the Delimiter-string. The following table shows a list of standard escape sequences:

\\	Backslash
\a	Alert
\b	Backspace
\f	Formfeed
\n	Newline
\r	Carriage return
\t	Horizontal tab

Table 4: Standard escape sequences

Furthermore the following escape sequences are recognized:

\x..	ASCII character as hex code
\0...	ASCII character as octal code
\v \V	Virtual key code (NOT IMPLEMENTED)

Table 5: Extended escape sequences

The hex code identifier "\x" must be followed by two digits (or characters 'a' to 'f' and 'A' to 'F' respectively). The octal code identifier "\0" (NOTE: backslash + zero!) must be followed by three octal digits (values 0..7).

E.g. you may specify the space character (decimal value 32) by "\x20" or "\0040".

5.2.4 String Replacement-Table

Section [TranslationTable] in the ini-file

You can add entries to the string replacement table using the syntax

```
searchstring=replacement
```

Thus every occurrence of "searchstring" in the input-stream will be replaced with "replacement" in the output-stream. E.g. you may specify the following table:

$1={CTRL-ENTER}	Replace "$1" with ctrl+enter
$2={TAB}	Replace "$2" with tabulator key
$3=Joe	Replace "$3" with "Joe"

Table 6: String replacement samples

> ► Make sure you have EnableControlSequ=Y for emulating keystrokes and EnableControl-Sequ=N for pure text replacement.

5.2.5 Control Sequence Settings

Section [ControlSequences] in the ini-file

EnableControlSequ=N	Use control sequences for the prefix, for the postfix and for the replacement table (right-hand value)?
ActivationDelay=100	Applications brought to front with the command {ACT:appname} will be given a response time of 100 ms before receiving data from *TWedgeCE*

Table 7: Control sequence parameterss

TWedgeCE recognizes a certain amount of control commands. They can be used to send key-strokes (e.g. CTRL+C for copy, CTRL+V for paste, TAB for setting the focus to the next input area etc.), to activate other program windows or to popup message boxes. Control sequences can be used in the prefix, in the postfix or in the right hand values of the string replacement table.

Control commands are enclosed in braces. E.g. you can send the key press sequence <Enter>, <Cursor Right> with the following control sequence:

```
{ENTER}{RIGHT}
```

► If EnableControlSequ=N the text “{ENTER}{RIGHT}” will be posted to your application and not the key press sequence.

For a list of valid control commands please refer to Appendix A.

5.2.6 Pre-Evaluation Expression

Section [PreEvaluation] in the ini-file

Expression=	Pre-evaluation expression (used to modify the acquired data).
Enable=N	Use pre-evaluation expression (y/n)?

Table 8: Pre-Evaluation parameters

The pre-evaluation expression is an expression, that can be used to modify the acquired data, before it is processed any further. The pre-evaluation is executed before the replacement of strings (see section 5.2.4) and before the execution of control commands (see 5.2.5). The prefix and the postfix are added to the modified input data afterwards.

Within the expression use the system variable “DATA” to reference to the acquired data string.

For a list of available functions, please refer to Appendix B.

Examples

Use the first five characters of the input only:

```
Expression=Left (DATA, 5)
```

Cut the first two characters of the input string:

```
Expression=Mid (DATA, 2, Len (DATA) - 2)
```

Prepend the prefix “pfx“ and append the suffix “sfx” to the input data:

```
Expression="pfx" + DATA + "sfx"
```

5.2.7 Settings for the Tray Menu

Section [TrayMenu] in the ini-file

Enabled=Y	Show the small program icon inside the taskbar (y/n)? (see figure 1 or 3, ❸)
Position=124	X-position of the icon in pixels. This setting allows you to position the icon within the taskbar. Depending on the operating system (PocketPC or Windows CE) only certain positions may work! We recommend testing various values!

Table 9: Tray menu parameters

5.2.8 Operation Mode

Section [Operation Mode] in the ini-file

KeyboardEmulation=Y	Emulate keystrokes thus other applications will receive data from *TWedgeCE*?

Table 10: Keyboard emulation parameters

► If disabled *TWedgeCE* will NOT send ANY data!

5.3 Example Configurations

5.3.1 No Newline, No Linefeed

Some barcode scanners may append a “newline” character and a “linefeed” character at the end of the scanned text. (In escape sequences this is “\r\n”). If you do not want these characters to be sent to your application you can define the sequence “\r\n” as input stream delimiter. Thus every data package will be cut with the occurrence of “\r\n”. If IncludeDelimiter is set to N the delimiter “\r\n” will not be included in the output.

```
Delimiter=\r\n
UseDelimiter=Y
IncludeDelimiter=N
```

5.3.2 MS Excel

To change to the next data cell in MS Pocket Excel after receiving data you can use the following line in the settings file:

```
Postfix={CTRL-ENTER}{DOWN}
```

The keystroke {CTRL-ENTER} fills the current cell with the received data. The keystroke {DOWN} selects the cell below for the next input.

5.3.3 Replacement Table

If you want to replace every occurrence of “$1” in the input stream with the content of your clipboard you can do this by the following statement. (It sends the paste command Ctrl+V every time “$1” occurs in the input string).

```
[TranslationTable]
$1={CTRL-V}
```

5.3.4 Extract Sub-String from the acquired Data

You can configure *TWedgeCE* to use a sub-string from the acquired input data only. To do this use the pre-evaluation expression. The following configuration will remove the first character and use the following 8 characters from the acquired data packet only:

```
[PreEvaluation]
Expression=MID(DATA,1,8)
Enable=Y
```

6 Examples of Use

6.1 Example 1

The input device provides the following data. Note: “\x0a\x0d” is the hex code for the newline linefeed combination (which could also be written as “\r\n”).

```
BC280229C00000000=00001T9\x0a\x0d
```

Task

Remove the first character "B", transmit the next 16 characters "C280229C00000000" as keystrokes, remove the end of the string "=00001T9\x0a\x0d". And finally press "enter".

Therefore you could use the following configuration:

```
[ComSettings]
;DataPackaging
Delimiter=\x0a\x0d
UseDelimiter=Y
IncludeDelimiter=N
Postfix={ENTER}

[PreEvaluation]
Expression=MID(DATA,1,16)
Enable=Y
```

The delimiter is used to identify the end of the input data and to crop the newline linefeed. The pre-evaluation extracts the relevant data from the input string. Finally, the postfix is used to send an “enter”.

7 Available Licenses

7.1 Retail License

Retail Licenses of *TWedgeCE* are available for

- **Single User**
  Installation on exactly one device
- **Workgroup**
  1-10 installations within one company on exactly one site (or subsidiary).
- **Office**
  1-100 installations within one company on exactly one site (or subsidiary).
- **Site**
  1-250 installations within one company on exactly one site (or subsidiary).
- **Enterprise**
  Unlimited number of installations worldwide (within one enterprise).

7.2 Demo License

A demo installation (for evaluation purposes) is available on http://www.tec-it.com.

7.3 Pricing

Please contact us for actual prices.

8 Contact and Support Information

TEC-IT Datenverarbeitung GmbH

Address:	Wagnerstr. 6 AT-4400 Steyr Austria/Europe
Phone:	+43 / (0)7252 / 72 72 0
Fax:	+43 / (0)7252 / 72 72 0 – 77
Email:	mailto:support@tec-it.com
Web:	http://www.tec-it.com

Appendix A: Control Commands

The following control commands are recognized by *TWedgeCE*. Please note that you have to put control commands in braces (e.g. {RETURN} for the return key)!

CANCEL	
BACKSPACE	Backspace key
TAB	Tabulator key
CLEAR	
ENTER	Enter key (keypad)
RETURN	Return key
SHIFT	Shift key
ESC	Escape key
SPACE	Space key
PAGE-UP	Page up key
PAGE-DOWN	Page down key
END	End key
HOME	Home key
LEFT	Cursor left key
UP	Cursor up key
RIGHT	Cursor right key
DOWN	Cursor down key
SELECT	
PRINT	Print key
EXECUTE	
SNAPSHOT	Print screen key
INS	Insert key
DEL	Delete key
HELP	
MUL	Multiply key (keypad)
ADD	Add key (keypad)
SEP	Separator key (keypad)
SUB	Subtract key (keypad)
DEC	Decimal key (keypad)
DIV	Divide key (keypad)
F1	F1 key
F2	F2 key
F3	F3 key
F4	F4 key
F5	F5 key
F6	F6 key
F7	F7 key
F8	F8 key
F9	F9 key

F10	F10 key
F11	F11 key
F12	F12 key
LMENU	Left alt key
RMENU	Right alt key (Alt Gr)
LWIN	Left windows key
RWIN	Right windows key
APPS	Application key (Microsoft Natural Keyboard)
ACT:window	Activate window named “window” and bring it to front
MSG:text	Displays “text” as message box

Table 11: TWedgeCE control commands

Toggle Keys

If you want to emulate the keystroke SHFIT+CTRL+ALT+S you have to use the control string {SHIFT-CTRL-ALT-S}

- **NOTE: ALWAYS USE THIS ORDER FOR TOGGLE KEYS!**
  (otherwise the control sequence will not be recognized correctly)
- 1. SHIFT
- 2. CTRL
- 3. ALT

Appendix B: Function Reference

B.1 Functions

Return	Function	Description
long	Abs («Number»)	Returns the absolute value of a number.
long	Asc («Text»)	Returns the ASCII value of a given character or of the first character of ”text”.
char	CDate («Text»)	Converts the string ”text“ to a date. Provides the current date (Now ()) if no conversion is possible.
double	CDbl («Expr»)	Converts any value to a double value (floating-point notation). The result is 0.00 when a conversion is not possible.
char	Chr («Number»)	Returns the corresponding character for the specified ASCII value ”Number“.
long	CLng («Expr»)	Converts any value into a whole number. If a conversion is not possible, the result is 0.
string	CStr («Expr»)	Converts a value into a text.
long	Day («Date»)	Determines the day of the month [1..31].
long	DayOfWeek («Date»)	Returns the day of the week of a specified date [1..7]. 1=Sunday, 2=Monday, ...
double	DayOfYear («Date»)	Returns the day of the year of a specified date [1..366].
double	Exp («Number»)	Returns the value $e^{"Number"}$, where e is the base of the natural logarithms.
double	Exp10 («Number»)	Returns the $10^{"Number"}$.
long	Find («Text», «SearchText», «nStart»)	Searches the string "Text" for "SearchText" starting from Position "nStart". Returns the position of the string or -1. The first character of a string is located at position 0.
long	FindReverse («Text», «SearchText», «nExclude»)	Searches the string "Text" for "SearchText" in reverse order excluding "nExclude" characters at the end. Returns the position of the string or -1. The first character of a string is found at position 0.
string	Format («Number», «Pattern»)	Formats "Number" according to the specified pattern string “Pattern“. Format placeholders: # digit or no value, 0 '0' or digit . decimal point , comma + - sign
double	Fract («Number»)	Returns the fractional unit
long	Hour («Date»)	The hour of a specified date [00..23].
string	IIf («Condition», «TrueExpr», «FalseExpr»)	Returns the value of «TrueExpr» if «Condition» is evaluated as (TRUE or not equal to 0) . Returns the value of «FalseExpr» if «Condition» is evaluated as (FALSE or equals 0).
long	IsEmpty («Text»)	Test whether the string "Text" is empty or not.
long	IsEven («Number»)	Returns TRUE if "Number" is even.
long	IsLastPage ()	Returns TRUE if the page being printed is the last page of the document.
bool	IsLeapYear («Date»)	Returns TRUE if the specified date occurs within a leap year.
long	IsOdd («Number»)	Returns TRUE if "Number" is uneven.
string	LastValue («DatafieldName»)	Returns the last value of a given data-field (the value of the data-field from the previous record).
string	Left («Text», «nLength»)	Returns the first characters "nLength” of a string.
long	Len («Text»)	Returns the length of the given string.
double	Log («Number»)	Returns the natural logarithm of "Number".
double	Log10 («Number»)	Returns the logarithm of “Number”.
string	Mid («Text»,	Returns the substring of string "Text" starting at position "nStart" with length

	«nStart», «nLenght»)	"nLength". The first character of a string is located at position 0.
long	Minute («Date»)	The minutes of a specified date/time [00..59].
long	«a» % «b»	Modulo operator: Remainder of the integer division a / b
long	Month («Date»)	The month of the specified date [1..12].
date	Now ()	Actual date and time.
double	Pow («Number», «Power»)	Returns the result of “Number” raised to the power of “Power”.
string	Replace («Text», «SearchText», «ReplaceText»)	Replaces each occurrence of "SearchText" in string "Text" with "ReplaceText".
string	Right («Text», «nLength»)	Returns the last characters "nLength" of a string.
double	Round («Number», «Precision»)	Returns “Number“ rounded using precision digits. If 0 is “Precision“ the result will be rounded to a whole number.
long	Second («Date»)	The second of a specified date/time [00..59].
double	Sqrt («Number»)	Returns the square root of “Number“.
long	SumOfDigits («Number»)	The sum of all digits of “Number“.
long	SumOfDigits1 («Number»)	Returns the one digit sum of all digits of “Number“.
string	ToLower («Text»)	Converts all character in the string “Text“ to lower case.
string	ToUpper («Text»)	Converts all character in the string “Text“ to upper case.
string	Trim («Text»)	Removes leading and trailing spaces.
string	TrimLeft («Text»)	Removes leading spaces.
string	TrimRight («Text»)	Removes trailing spaces.
double	Value («Text»)	Converts “Text“ to a double value.
long	WeekOfYear («Date»)	The calendar week of a specified date/time [1..52].
long	Year («Date»)	The year of the specified date/time.

Table 12: Functions

B.2 Constants

Function	Description
False	Logical value FALSE. This value is usually the result of a condition. If the condition is not fulfilled the resulting value is FALSE.
True	Logical value TRUE. This value is usually the result of a condition. If the condition is fulfilled the resulting value is TRUE.
"\n"	Linefeed.

Table 13: Constants