TWedge

Data Acquisition Software

Version 1.2

User Documentation

20 February 2006

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

2 Disclaimer

The actual version of this product (document) is available as is. TEC-IT declines all warranties which goes beyond applicable rights. The licensee (or reader) bears all risks that might take place during the use of the system (the documentation). TEC-IT and its contractual partner cannot be penalized for direct and indirect damages or losses (this includes non-restrictive, damages through loss of revenues, constriction in the exercise of business, loss of business information or any kind of commercial loss), which is caused by use or inability to use the product (documentation), although the possibility of such damage was pointed out by TEC-IT.

TEC-IT Datenverarbeitung GmbH
Wagnerstr. 6

A-4400 Austria
t.: +43 (0)7252 72720
f.: +43 (0)7252 72720 77
http://www.tec-it.com

3 Introduction

3.1 About

TWedge is an application designed to acquire data over a serial interface or over TCP/IP. You can capture data from any device and forward it to any target application without a single line of program code.

3.2 Communication

The following interfaces are supported:

- Serial Interface (all communication parameters are adjustable)
- TCP/IP (in client mode)

3.3 Keyboard Simulation

TWedge allows you to translate captured data into keystroke sequences (so called “Key Wedge Mode”). Thus it is possible to input data from external devices into any application without switching application windows.– Single keystrokes can be sent as well as keyboard shortcuts, customizable Control Sequences offer you flexible automatization functions (– e.g. the required input field can be selected automatically, or applications may be activated on demand).

Keystroke simulation is a very efficient way to integrate data acquisition features into existing applications. No modification or redesign is required. Even ERP-systems like SAP® R/3® and all type of legacy applications benefit from this slim solution.

3.4 File Logging

The received data can be automatically written to output files (log files) in given time intervals. File name and time interval are adjustable.

3.5 Automatic Initialization

On start-up the following steps can be executed automatically:

- Load the required program settings (including the connection parameters)
- Minimize *TWedge* to the task bar (or to the system tray)
- Start communication

3.6 Hot-Key Support

On demand, *TWedge* may be configured to monitor an adjustable hot-key. Whenever the user presses this hot-key a character sequence is sent to the connected device. This feature enables automatic data acquisition on demand.

4 Starting TWedge

4.1 Start TWedge with Default Settings

When *TWedge* is started from the Start Menu (without parameter), the default program settings will be loaded. Nevertheless configuration files can be loaded as described in 5.2.5.

4.2 Start TWedge with Customized Settings

If you want to start TWedge with a previously stored configuration, you can

- Pass the configuration file as command line parameter
- Double-click on the configuration (.TWI) file

In both cases the following things will happen:

1. *TWedge* will load the .TWI-file.
2. All settings, which were loaded from the .TWI-file, will be applied.
3. Data Acquisition will start automatically.
   Depending on the configuration, a log-file will be generated and/or keystroke simulation will be enabled (Key Wedge mode).

> ➢ Please keep in mind that the communication parameters for your scanner (or data acquisition device) should match the settings in *TWedge;* otherwise data may not be received (or it may not be received correctly). – Setting up the communication parameters is described in 6.1.1.

4.3 Command Line Parameters

If the path, the file name and the extension of a configuration file is passed on the command line (or as parameter in the target field of a shortcut – see 4.4), *TWedge* will start with the settings, which were loaded from this file.

Examples:

```
TWedge1.exe TWedgeCOM1.TWI
```

```
"C:\Program Files\TEC-IT\TWedge1\TWedge1.exe" C:\Scale.TWI
```

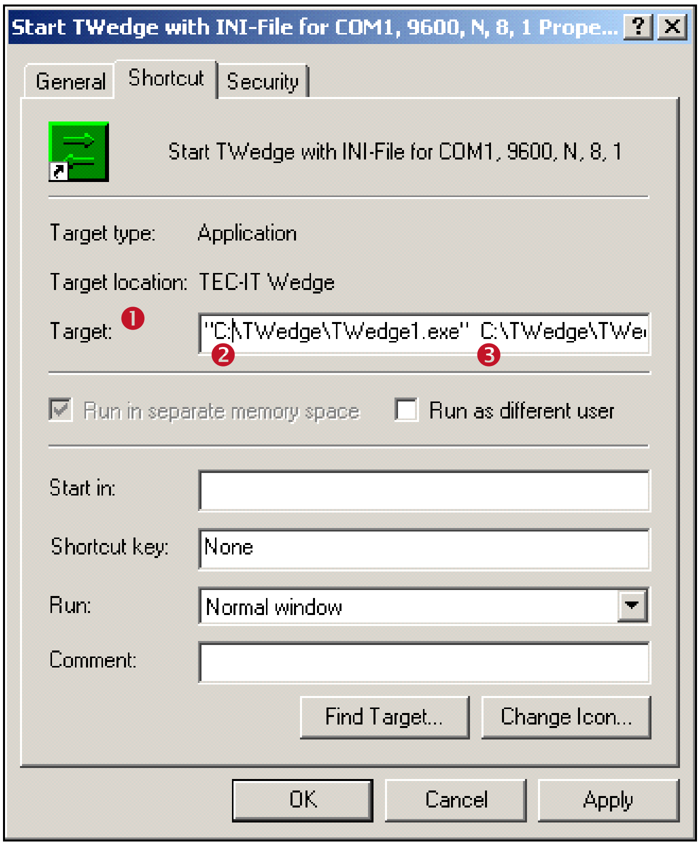
The first sample requires the files “TWedge.exe” and “TWedgeCOM1.TWI” to be present in the same directory where the command is executed. In the second sample the whole path to each of the files is specified. Thus the command can be run from any directory on your computer. *“TWedgeCOM1.TWI”* and *“Scale.TWI”* are both configuration files for *TWedge*.

> ➢ If there are any spaces in the file path (as in “C:\Program Files\...”) the whole path must be put under quotes. (Otherwise Windows will try to start “C:\Program” with the parameter “Files\TEC-IT\...” a.s.o.).
>
> ➢ If the configuration file contained license information, *TWedge* will be licensed automatically with the information given in this file. For more information, please refer to 8.1.1.

4.4 Creating Shortcuts

If you want to create a shortcut please follow the steps below:

Under Windows® XP open the Windows Explorer, go to the *File* menu and click *New* ► *Shortcut*. A wizard will appear. This wizard lets you select the target and the name for the shrotcut. Choose “TWedge1.exe” as target. (This file can be found under “C:\Program Files\TEC-IT\TWedge1\“ by default). The name can be choosen freely. To edit the shortcut right-click the file and select *properties.* The following dialog will appear.



Start TWedge with INI-File for COM1, 9600, N, 8, 1 Prope...

Start TWedge with INI-File for COM1, 9600, N, 8, 1

Target type: Application

Target location: TEC-IT Wedge

Target: "C:\TWedge\TWedge1.exe" C:\TWedge\TWe

☑ Run in separate memory space □ Run as different user

Start in:

Shortcut key: None

Run: Normal window

Comment:

Figure 1: Shortcut Dialog

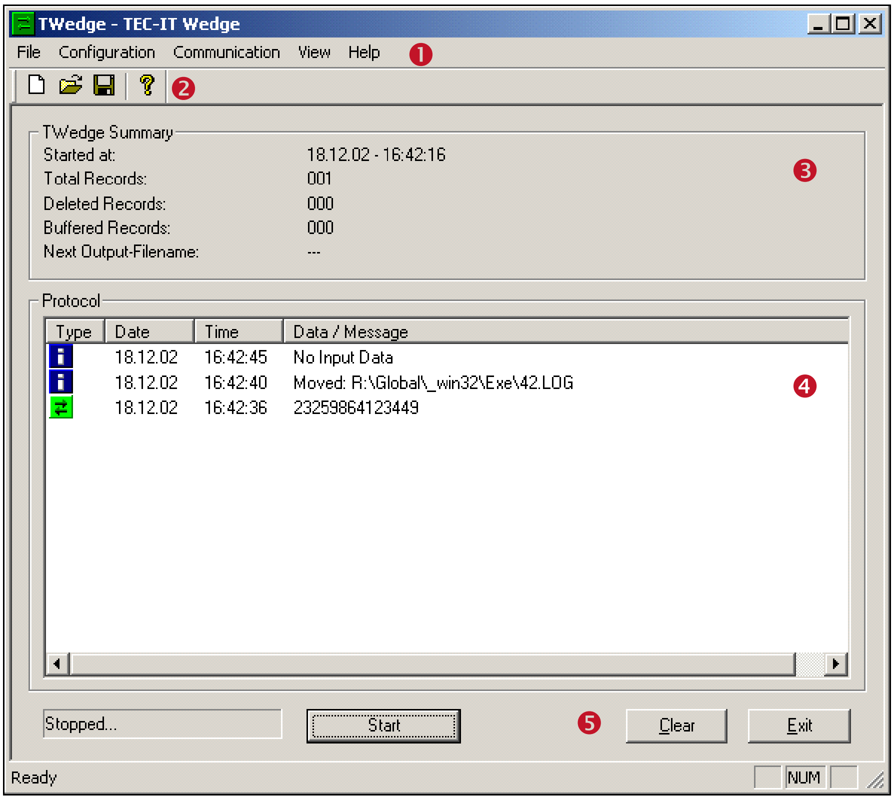
Field ❶ must contain the name and the path to the executable. Additionally a command line parameter can be specified.

In this example TWedge1.exe is located under “C:\Twedge\”, thus the target application is "C:\TWedge\TWedge1.exe” (see ❷).
Separated by a space character the executable is followed by the path and the name of the configuration file. – In this case “C:\TWedge\ TWedge.TWI” (see ❸).

After adjusting the shortcut click on the *OK* button to confirm the settings.

5 User Interface

5.1 Main Window



TWedge Summary

Started at:	18.12.02 - 16:42:16
Total Records:	001
Deleted Records:	000
Buffered Records:	000
Next Output-Filename:	---

Protocol

Type	Date	Time	Data / Message
i	18.12.02	16:42:45	No Input Data
i	18.12.02	16:42:40	Moved: R:\Global_win32\Exe\42.LOG
⇄	18.12.02	16:42:36	23259864123449

Stopped...

Ready NUM

Figure 2: Main Window

The main window is split into the following areas:

❶ Menu.
❷ Toolbar.
❸ General Information and File Logging Info.
❹ Log window.
❺ Main Controls.

5.2 General Operations

5.2.1 Configuring TWedge

To adjust the output file and the Key Wedge Mode, go to the menu ❶ and select *Configuration.* This submenu lets you bring up various configuration dialogs. (For detailed information please refer to chapter 6).

5.2.2 Saving your Settings

To save your current settings select *File* ► *Save* (or *Save As)* from the menu ❶, or click on the 💾 symbol in the tool bar ❷. A dialog window will appear. – In this dialog you can assign a file name and you can select the location where you want so save your settings file. The extension .TWI will be added automatically. (*TWedge* configuration files are identified by the .TWI extension).

5.2.3 Receiving Data

To start receiving data click the *Start* button in the main controls ❺. (As soon as the connection is started, this button will change its caption to “Stop”, indicating that the communication with the attached device is now started). *TWedge* starts receiving data with respect to the previously applied settings.

➢ *TWedge* can also be started automatically on start-up – for details please refer to chapter 4.2.

5.2.4 End Data Transmission

To end data transmission click on the *Stop* button in ❺ (which will now change its caption back to “Start” again). If *data logging* was enabled, the buffered data will be moved to the designated output file.

5.2.5 Loading a Saved Configuration

To load a saved configuration, select *Open* from the menu ❶ or click on the 📂 symbol in the tool bar ❷. A dialog will appear. – In this dialog you can browse for previously saved configuration files (*.TWI). Select the desired file and click on the *Open* button to load the configuration.
Data acquisition will not start automatically. Click on the *Start* button to start waiting for data.

5.3 TWedge Summary ❸

This section displays the following information:

Started at	The starting date and time of *TWedge*.
Total Records	The total amount of data strings which were read since the program start-up.
Deleted Records	The number of discarded records.
Buffered Records	The number of records, which were read within the specified time intervals.
Next Output-Filename	The next output file name used for storing data.

5.4 Log Window ❹

All events are listed in this window. The entries are structured as follows:

Type	Shows the type of the corresponding entry. ⇄ … Data ℹ … Information ❗ … Error Message
Date	Contains the date on which the log entry was generated.
Time	Contains the time at which the log entry was generated.
Data / Message	If data has been read, it shows the received data. Otherwise a message with additional information will be displayed.

A maximum of 1000 entries can be displayed in this screen. Any log entries exceeding this amount will be automatically discarded. (This has no influence/effect on log files).

5.5 Main Controls ❺

The main controls are located below the log window:

Start / Stop	Activates or deactivates data acquisition from the selected interface.
Clear	Deletes all entries in the log window (no buffered data will be lost).
Close	Exits *TWedge*. An active session will be closed after user confirmation only.

5.6 Menu ❶

5.6.1 File

New	All parameters will be reset to default values.
Open	Opens a configuration file. These files have the extension .TWI and contain saved program settings. TWI-files are ASCII-files and can be edited with any common text editor.
Save and Save As	The program settings will be saved in a file with the extension .TWI.
Close	Exits *TWedge*. An active session will be closed after user confirmation only.

5.6.2 Configuration

Interface	Opens a dialog in which the interface properties can be set.
Pre-Evaluation Settings	Pre-evaluation of the input data. (Extract the first n bytes, add a prefix/postfix, a.s.o.).
Key Wedge Settings	Generate keystrokes from received data and adjust control sequences.
Log File Settings	Configure Data Logging (time interval, file name, etc.).
Hot-Key	Hot-Key configuration. (The hot key allows sending data).
Start-Up Options	This dialog lets you adjust the *TWedge* start-up options (minimized, system tray).

5.6.3 Communication

Start / Stop	Start/Stop data acquisition.
Restart	Closes the interface, stops data acquisition and starts again.

5.6.4 View

Toolbar	Displays or hides the Toolbar.
Status Bar	Displays or hides the Status bar.

5.6.5 Help

About TWedge	Detailed version information and copyright details.
License the product	To license your version (= deactivate the Demo-Mode), a license key must be entered inside this dialog. The license key will be stored permanently in the registry. Installing or moving the software to another computer system requires re-entering the license key. To avoid this you can edit your .TWI settings file and enter the key there (see 8.1.1). Your computer's System ID will be be displayed in this dialog (needed for ordering a “Single” license).
Go to www.tec-it.com	If you want to learn more about our products, please visit us at http://www.tec-it.com. On this page you will find the latest product versions and user documentations.

6 Configuration

The *Configuration* menu gives you access to different TWedge settings. All of these settings can be saved in a configuration file and are available for being restored later on. *TWedge* Configuration files use the file extension .TWI.

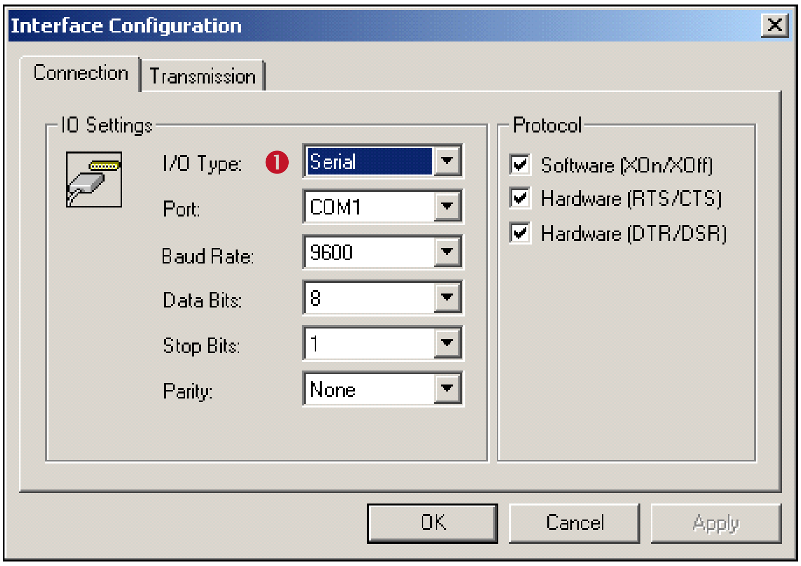
6.1 Interface

The Interface Configuration dialog consists of two different tabs:

- Connection
- Transmission

6.1.1 Connection

In the Connection tab you can set the interface properties for the connected device.



Interface Configuration

IO Settings

I/O Type: ❶ Serial

Port: COM1

Baud Rate: 9600

Data Bits: 8

Stop Bits: 1

Parity: None

Protocol

☑ Software (XOn/XOff)

☑ Hardware (RTS/CTS)

☑ Hardware (DTR/DSR)

Figure 3: Interface Settings – Serial Interface

Two different *I/O types* (❶) are available: you can select either “Serial” or “TCP”. – Each of these two interfaces has different settings:

6.1.1.1 Communication settings for the Serial (RS-232) interface

I/O Type	Select the entry “Serial” for serial communication (over the RS-232 interface).
Port	This is the interface to which the scanner or device is connected. (COM1, COM2, COM3, COM4). For addressing additional COM Ports (> COM9) please contact us.
Baud Rate	Sets the transmission rate, 9600, …, 115200, …, 256000)
Data Bits	Number of bits.
Stop Bits	Number of stop bits.
Parity	Sets the parity.

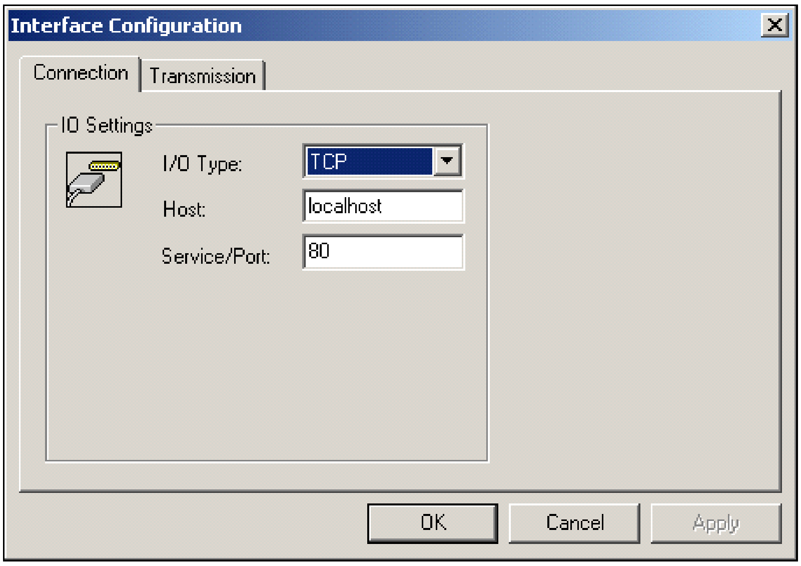
The following handshake modes can be configured:

Software (XOn/XOff)	Uses XOn/XOff Software Handshake.
Hardware (RTS/CTS)	Uses RTS/CTS Hardware Handshake.
Hardware (DTR/DSR)	Uses DTR/DSR Hardware Handshake.

If no handshake is required simply leave all checkboxes unmarked (not recommended).

6.1.1.2 Communication settings for TCP/IP:

When the *I/O Type* is set to "TCP" you can connect to a host computer (or device) in TCP/IP mode (*TWedge* always connects in client mode). The following settings will be available:



Interface Configuration

Connection | Transmission

IO Settings

I/O Type: TCP

Host: localhost

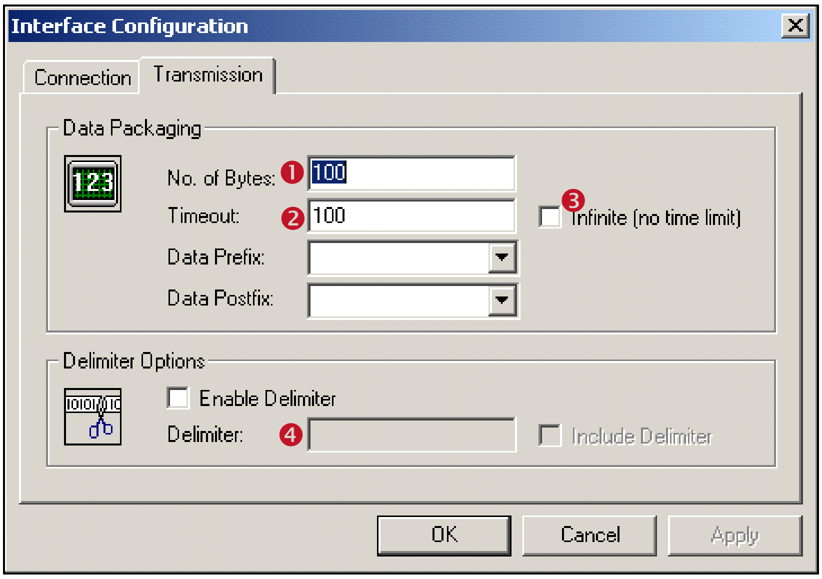
Service/Port: 80

Figure 4: Interface Settings – TCP/IP

I/O Type	Select the entry "TCP" for communication over TCP/IP. This can be used for connecting to a host by either specifying its IP address or its hostname.
Host	The Hostname (e.g. www.tec-it.com) or the IP address (e.g. 127.0.0.1) of the TCP/IP Server.
Service/Port	Enter the service name (e.g. http) or the port number (e.g. 80).

6.1.2 Transmission

TWedge reads data asynchronously. This means the received data is being collected and will only be processed further (simulated as keystrokes) if certain conditions come true. – i.e. if a given number of bytes (characters) has been received or if a specified time limit has been elapsed. Therefore *TWedge* uses data packages.



Interface Configuration

Data Packaging

No. of Bytes: ❶ 100

Timeout: ❷ 100

❸ □ Infinite (no time limit)

Data Prefix:

Data Postfix:

Delimiter Options

□ Enable Delimiter

Delimiter: ❹

□ Include Delimiter

Figure 5: Transmission Settings

6.1.2.1 Data Packaging

The received data stream will be divided into packages according to the settings for size (❶), timeout (❷ and ❸) and data content (= match with a given delimiter string ❹). Every time a new data package is complete further processing of the data package will be started. The following conditions are used for splitting the accumulated data into packages:

No. of Bytes	The maximum number of bytes per package. One package may never exceed the “No. Of Bytes” size limit.
Timeout	All data received within “Timeout” milliseconds will be accumulated into one package. When Timeout has elapsed, the package is complete. Only a maximum number of bytes (property "No. Of Bytes") will be included into one package. Timeout begins to run as soon as the first (new) byte of the data transmission is received. A <u>negative value</u> in the timeout field is used to indicate, that the given time must elapse without receiving any data. Then the previously received bytes will be put into a package. This may be useful in conjunction with scales, which are sending data continuously until the weight has stabilized. Please consider, that the negative timeout works in combination with a delimiter only: The data between the last two delimiters will be used for further processing.
Infinite	When activated, there is no time limit for creating packages. Only the value “No. of Bytes” (as well as an optional delimiter) will be used for splitting the incoming data.
Data Prefix	The Data Prefix will be inserted <u>before</u> each data package. Thus you have the possibility to prepend a constant string or to execute a predefined Action sequence before the processing of the received data. You can use the so-called "Control Sequences" (see 6.3.1) within the data prefix. Thus you can select a certain input window or you can select a certain input field before simulating the received data as keystroke sequence. The listbox offers some predefined prefixes, but – for use as Control Sequence – you can also enter a self-defined character string (= one of those strings, which you have defined in the Key Wedge Settings as Trigger-Sequence – see 6.3.1).
Data Postfix	The Data Postfix will be inserted <u>after</u> each data package. Despite that it can be used in the same way as the Data Prefix.

<u>Sample:</u> If your barcode scanner reads barcodes containing 14 characters each, and if the scanner transmits with a Baud Rate of 9600 bits/sec (~ 950 bytes/sec) you could adjust the "No. of Bytes" to 20 and set the Timeout to 150 ms to be on the safe side.

6.1.2.2 Delimiter Options

The delimiter is a character (or a character sequence), which is used for splitting the incoming data depending on its content. With each occurance of the delimiter in the incoming data stream a new package will be created. The delimiter string can be specified in hexadecimal ASCII coding or in byte coding.

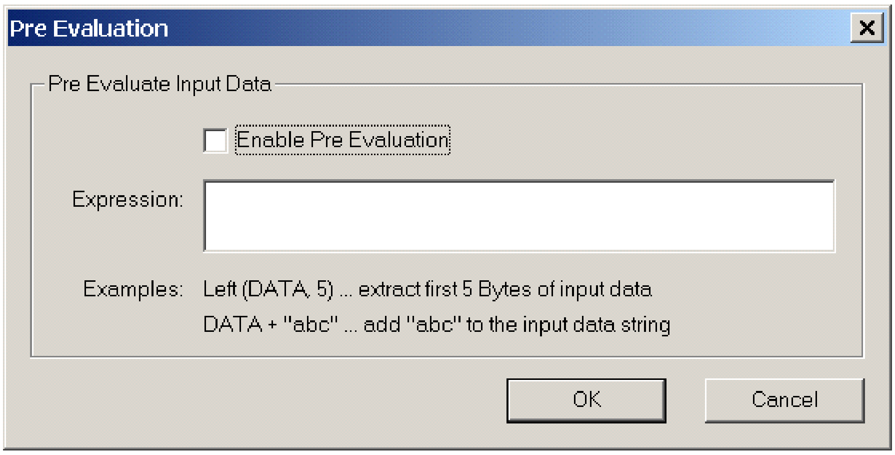
For example you could use Carriage Return / Line Feed as delimiter characters in conjunction with a barcode reader. Thus the values for the properties “No. of Bytes” and “Timeout” can be increased, because data packaging (and therefore data processing) will be triggered by the delimiter prior to the other two conditions.

Enable Delimiter	Specifies whether delimiters should be used or not. If enabled, each occurance of the delimiter string will split the received data into packages.
Delimiter	This string will be used as delimiter. The received data is permanently scanned for this string. Each time the complete Delimiter has been detected, a package is generated from the data received thus far. Afterwards the data will be processed (= forwarded to keystroke simulation and/or data logging). These are common delimiter strings: ▪ \x0d …Hex code for Carriage Return (ASCII) ▪ \x0a …Hex code for Line Feed (ASCII) ▪ \x0d\x0a …Carriage Return + Line Feed (often used with barcode readers)
Include Delimiter	This flag specifies, if the Delimiter should remain part of the data package or if it should be omitted (stripped off).

- You need a delimiter when using the negative timeout feature (see 6.1.2.1). This feature may be used for scales, which are sending data continuously until the weight has stabilized.

6.2 Pre-Evaluation

The Pre-Evaluation is the first processing step in *TWedge*. It allows you to modify all incoming data before it is processed any further. The input data can be modified by using an editable expression. This expression will be evaluated each time input data is received.



□ Enable Pre Evaluation

Examples: Left (DATA, 5) ... extract first 5 Bytes of input data
DATA + "abc" ... add "abc" to the input data string

Figure 6: Pre-Evaluation Settings

Enable Pre-Evaluation	Enables/Disables the pre-evaluation.
Expression	Here you can enter the expression, which shall be used for modification of the input data. Examples for pre-evaluation expressions can be found below.

Expressions are built upon these three entities:

- **Variables**
  The variable DATA identifies the input string which was acquired through the attached hardware.
- **Operators**
  You can use the operator + to concatinate two strings.
- **Functions**
  A list of available functions (e.g. string manipulation functions) can be found in chapter 8.4.1.

6.2.1 Examples

Use the first five characters of the input only:

```
Left (DATA, 5)
```

Cut the first two characters of the input string:

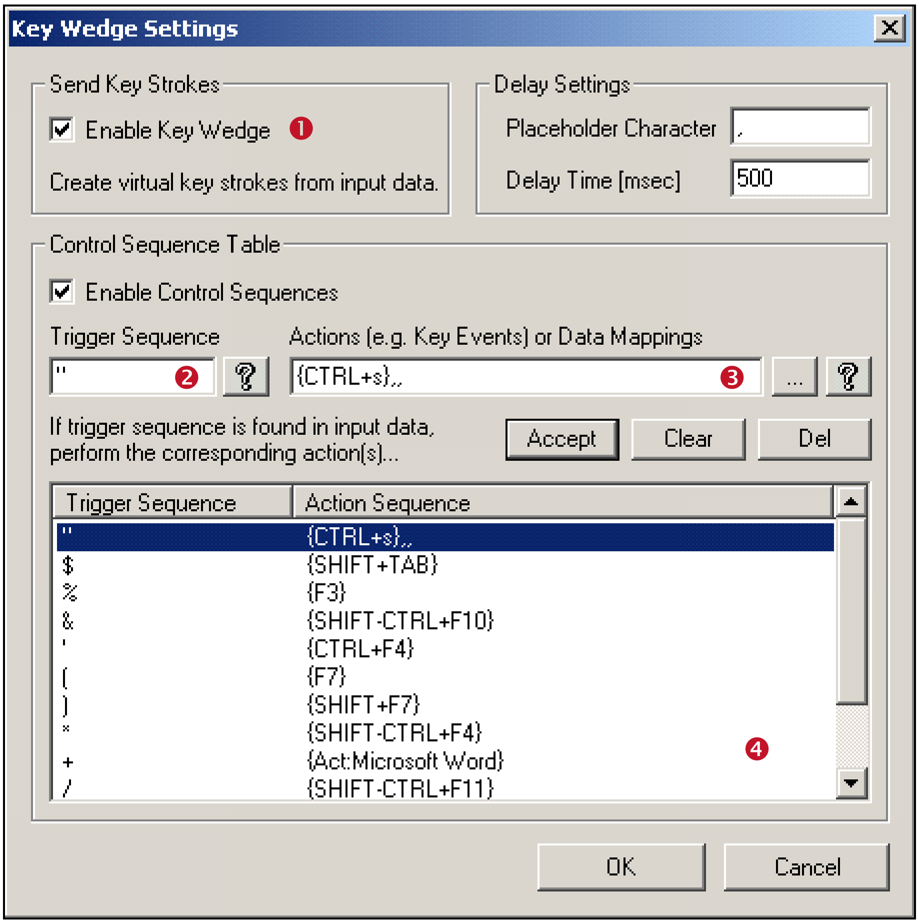
```
Mid (DATA, 2, Len (DATA) - 2)
```

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

```
"pfx" + DATA + "sfx"
```

6.3 Key Wedge Settings (Control Sequences)

In this dialog the Key Wedge (= keystroke simulation) can be enabled or disabled (❶). The Key Wedge is required if you want to use Control Sequences.



☑ Enable Key Wedge

Create virtual key strokes from input data.

Placeholder Character: ,

Delay Time [msec]: 500

☑ Enable Control Sequences

Trigger Sequence: "

Actions (e.g. Key Events) or Data Mappings: {CTRL+s},,

If trigger sequence is found in input data, perform the corresponding action(s)...

Trigger Sequence	Action Sequence
"	{CTRL+s},,
$	{SHIFT+TAB}
%	{F3}
&	{SHIFT-CTRL+F10}
'	{CTRL+F4}
(	{F7}
)	{SHIFT+F7}
*	{SHIFT-CTRL+F4}
+	{Act:Microsoft Word}
/	{SHIFT-CTRL+F11}

Figure 7: Key Wedge Settings Dialog

Enable Key Wedge	If enabled, the received data will be simulated as keystrokes (so-called Key Wedge Mode).
Enable Control Sequences	Enables/Disables the use (decoding) of Control Sequences.
Delay / Placeholder	Definable placeholder. If this character occurs in the "action sequence", a break will be inserted. This can be used to give the target application some time before sending the next command. (e.g. after an {Alt-S} for saving the data).
Delay / Time	This is the delay in milliseconds, which will be executed, if the Delay-Placeholder occurs in the Action-Sequence. (Input 1000 = 1 Second).
Trigger Sequence	Input field to enter (or modify) a Trigger Sequence.
Actions	Input field to enter (or modify) the Action Command that should be executed when the Trigger Sequence is found.
Accept	Transfers the values from ❷ and ❸ to table ❹. To edit an existing control sequence select the item in the table (mouse or cursor), edit the value and click "Accept".
Clear	Clears the input boxes. Thus you can enter a new Trigger/Action Sequence.
Del	Deletes the selected entry from the table.
[...]	Opens the Action Sequence Wizard – here you can generate a key combination or a command that will automatically be inserted in the “Action” field.
[?]	Opens an information window.

6.3.1 Control Sequences

A *Control Sequence* consists of a *Trigger Sequence* (❷) and an *Action Sequence* (❸). If a Trigger-Sequences is detected within the input data (or within the prefix data or postfix data – see 6.1.2.1), the assigned Action Sequence will be executed. (An Action Sequence may be pressing a keyboard shortcut or activating a window, e.g.). Multiple Control Sequences (consisting of Trigger and Action) can be specified and inserted into the control sequence table (❹). Thus different actions can be triggered upon the occurance of different trigger strings.

Control Sequences can be used for automating the data input from a barcode reader. You can insert a special trigger-string into the barcode itself or you can insert a trigger string in the pre-fix/postfix data – thus the corresponding action command(s) will be inserted preceding (or succeeding) the actual data. This makes it possible to select an input field (or to bring up an application window) before pasting the data.

6.3.2 Action Sequences

Action sequences can contain keyboard commands as well as characters, numbers, or any other printable character. The following key identifiers may be used for non-printable keys.

CANCEL	HOME	F1	MUL
BACKSPACE	END	F2	ADD
TAB	PAGE-UP	F3	SEP
CLEAR	PAGE-DOWN	F4	SUB
ENTER	LEFT	F5	DEC
RETURN	RIGHT	F6	DIV
SHIFT	UP	F7	LMENU
ESC	DOWN	F8	RMENU
SPACE	PRINT	F9	LWIN
SELECT	INS	F10	RWIN
EXECUTE	DEL	F11	APPS
HELP		F12	

Table 1: Key identifiers for non-printable keys

Example:

if you want to press F10 and then enter the character string ”ABC“ you will have to enter the Action Sequence “{F10}ABC”.

- ➢ Simple string-replacement can be done by using character strings for the Trigger Sequence and for the Action Sequence.

6.3.2.1 Syntax

All keyboard commands have to be enclosed in braces. – e.g. {RETURN} or {CTRL-s}. The braces separate a command from a normal character.

Commands can also be executed consecutively, e.g.

```
{F10}{SHIFT-RETURN}
```

6.3.2.2 Action Command Wizard

To ensure that the right syntax is used we recommend using the Action Command Wizard. To enter the wizard, please press the small *[…]* button beside the input field. The wizard allows you to insert key stroke commands (e.g. { SHIFT-CTRL- s}) as well as the action commands “window activation” and “message box” (see 6.3.2.4 and 6.3.2.5).

6.3.2.3 *Key Combinations*

When pressing more than one key simultaneously, the subsequent format has to be used (in exact this order).

```
{ [SHIFT-] [CTRL-] [ALT-] Key }
```

- ➢ e.g. {SHIFT-CTRL-F10} is OK, however {CTRL+SHIFT-F10} is incorrect, instead of STRG (German) always use the English name CTRL!

6.3.2.4 *Window Activation*

The command “Act:” followed by the window caption sets the focus on the window with the given caption. Minimized windows and windows in the background are brought to foreground (activated). Thus you can make sure, that the data is entered inside the right window.

Syntax:

```
{Act: Window Caption}
```

- ➢ To determine the caption of a window, press the combination ALT+TAB. The captions of all open windows can be displayed in this way. (Hold the Alt key and successively press Tab to step through all open windows).

Example:

```
{Act:Microsoft Word}{RETURN}
```

…selects Microsoft Word as active window and inserts a Carriage Return.

- ➢ If the destination Window is inoperative (if the application is not started) the subsequent commands in this Action sequence will not be executed; however the data is entered in the currently focused window.

6.3.2.5 *Message Box*

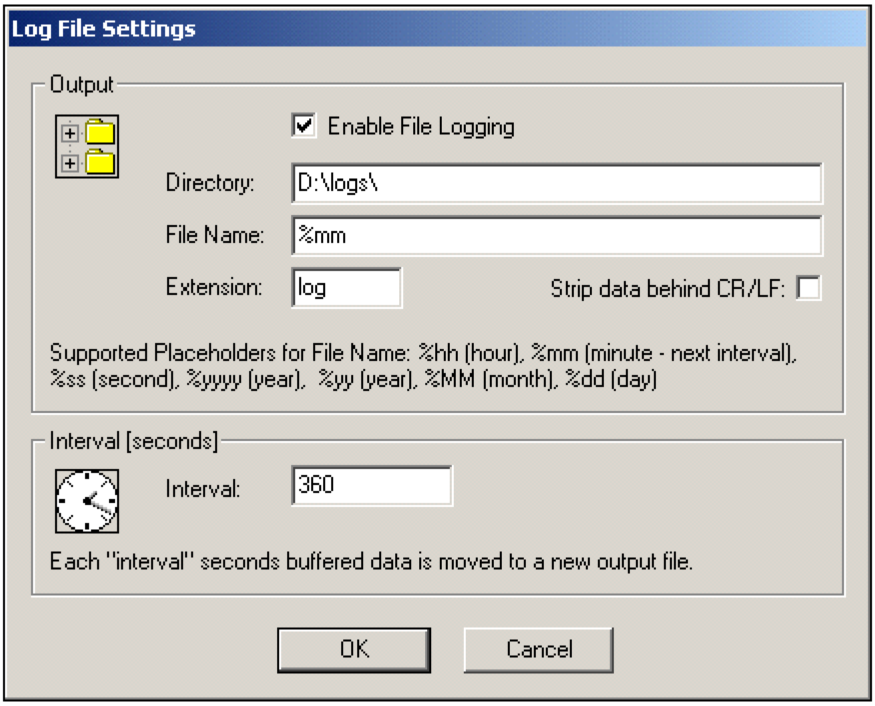
A Message-Box can be displayed using the “Msg:” command. – The keyboard-emulation will wait until the message has been acknowledged (OK Button).

Syntax:

```
{Msg:My Messages}
```

6.4 File Settings

In the following dialog the output file settings (Logging) can be adjusted.



Log File Settings

Output

☑ Enable File Logging

Directory: D:\logs\

File Name: %mm

Extension: log

Strip data behind CR/LF: □

Supported Placeholders for File Name: %hh (hour), %mm (minute - next interval), %ss (second), %yyyy (year), %yy (year), %MM (month), %dd (day)

Interval [seconds]

Interval: 360

Each "interval" seconds buffered data is moved to a new output file.

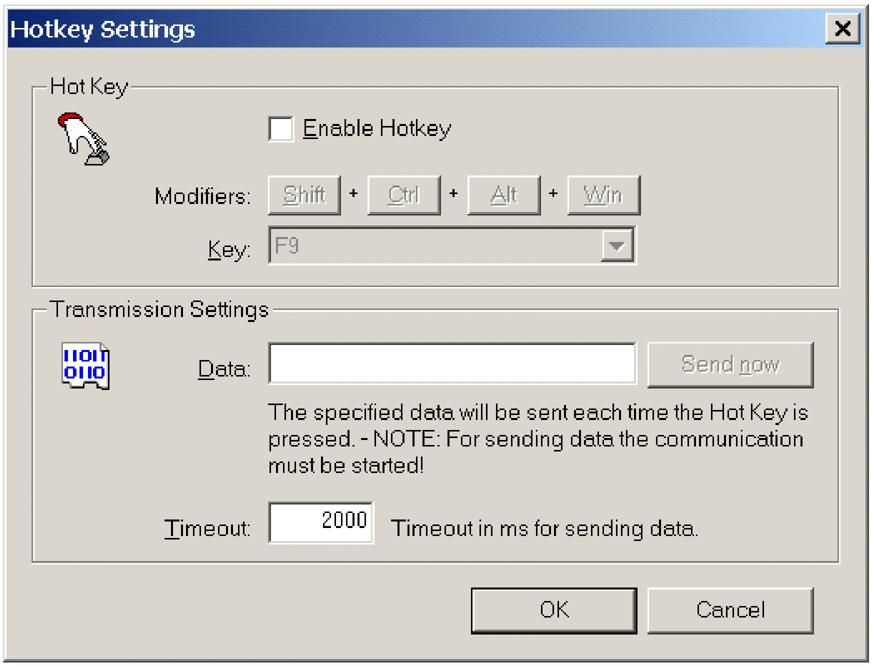
Figure 8: Log File settings

Directory	Path to the output files.
File Name	Name for the output file. The file name may contain placeholders. Placeholders: %hh - Hour, in which the data was generated %mm - Minute, in which the data was generated %ss - Second, in which the data was generated %yyyy - Year, in which the data was generated (4-digit) %yy - Year, in which the data was generated (2-digit) %MM - Month, in which the data was generated %dd - Day, on which the data was generated (For an example see 8.2)
Extension	File name extension.
Strip data behind CR/LF	If checked, all data following a carriage return (CR) + line feed (LF) will be removed.
Interval	Time in seconds, in which the buffered data is moved to the specified output file. First the data is saved in an internal buffer. The “interval” regulates the time period, after which the data is written to the log file. At the end of each time interval a new log file is created.

6.5 Hot-Key

The hot-key allows you to send a (pre-defined) string to the attached device.

Provided the communication is started, the string will be sent to the interface each time the hot-key is pressed. The hot-key may be used for devices, which need to receive a certain control string before they start data transmission.



Hotkey Settings

Hot Key

☐ Enable Hotkey

Modifiers: Shift + Ctrl + Alt + Win

Key: F9

Transmission Settings

Data:

The specified data will be sent each time the Hot Key is pressed. - NOTE: For sending data the communication must be started!

Timeout: 2000 Timeout in ms for sending data.

Figure 9: Hot-Key Settings

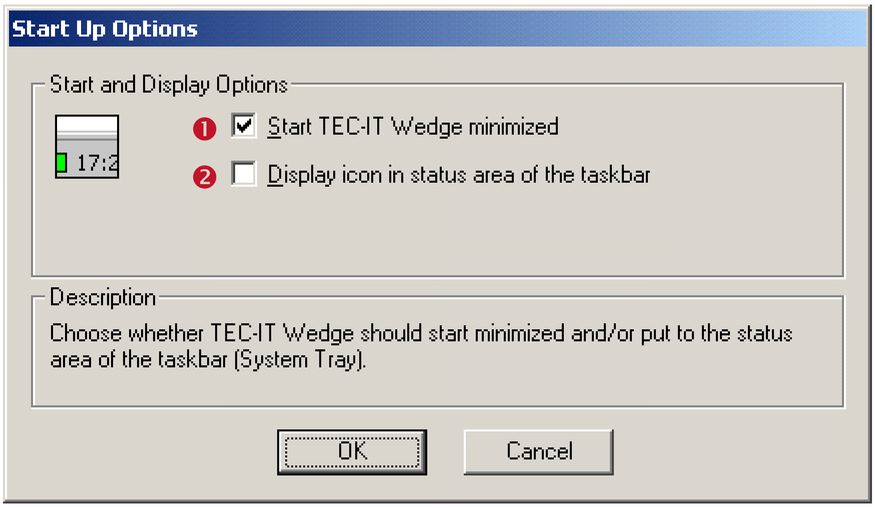
Enable Hotkey	Activates/Deactivates the hot-key.
Modifiers	Key modifiers for the shortcut. (Multiple selections are possible).
Key	Key selection. To use the shortcut Ctrl+F9 as hot-key, you have to select the "F9" key. Additionally the button "Ctrl" (from the Modifiers) must be pressed. The hot-key function must be enabled in order to change the keyboard shortcut!
Data	Input field for the transmission data. This data will be sent when the hot-key is pressed. The usage of escape sequences (like "\n") is possible. Hint: If the communication is started, the transmission can be triggered by pushing the "Send now" button, too. This might be useful for testing purposes.
Timeout	Timeout for the transmission (in milliseconds). If the addressed device does not respond within the given time period, the transmission attempt will be cancelled and the user will be informed.

- Certain shortcuts may not be working (e.g. Win+PRINT, Win+TAB, …). Please try different shortcuts instead!

6.6 Start-Up Options

In this dialog the start-up options for *TWedge* can be set.

If “Start TWedge minimized” (❶) is selected, the program will be minimized upon startup. Additionally you can make *TWedge* minimize to the system tray (❷) (instead of minimizing to the taskbar). If selected, *TWedge* will be shown as an icon in the status area of the taskbar only (next to the clock). The color of the icon will show the current connection status.



Start Up Options

Start and Display Options

❶ ☑ Start TEC-IT Wedge minimized

❷ □ Display icon in status area of the taskbar

Description

Choose whether TEC-IT Wedge should start minimized and/or put to the status area of the taskbar (System Tray).

Figure 10: Start-Up Options

Start TWedge minimized	Starts TWedge minimized.
Display icon in status area of the taskbar	Starts TWedge minimized in the status area of the taskbar.

7 Licensing

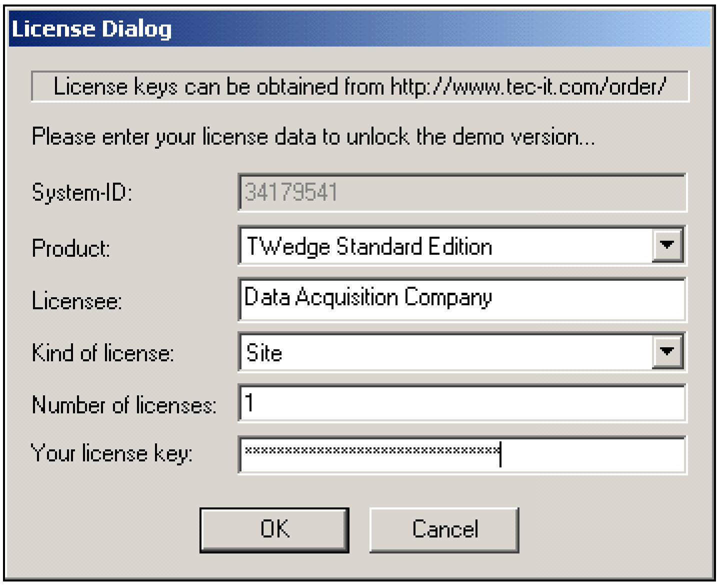
7.1 License Types

In the licensed version of *TWedge* the random insertion of demo data will be deactivated. The following license types are available:

Single	Installation on exactly one computer.
Workgroup	1 to 10 installations within one company on exactly one site (or subsidiary).
Office	1 to 100 installations within one company on exactly one site (or subsidiary).
Site	1 to 250 installations within one company on exactly one site (or subsidiary).
Enterprise	Unlimited installations within one enterprise.

7.2 Applying the License Key

The license information can be entered in the following dialog. You can bring up this dialog by choosing *Help* ► *License the product…* from the menu.


License Dialog

License keys can be obtained from http://www.tec-it.com/order/

Please enter your license data to unlock the demo version...

System-ID: 34179541

Product: TWedge Standard Edition

Licensee: Data Acquisition Company

Kind of license: Site

Number of licenses: 1

Your license key: ********************************

Figure 11: License Dialog

System-ID	The System-ID of your computer.
Product	Name of the program to be licensed (No selection necessary).
Licensee	Name of the license holder.
Kind of license	License type: ▪ Single (Single license) ▪ Site (Site license for one company location) – Use this selection for the following license types: “Workgroup”, “Office”, “Site”, “Enterprise”.
Number of licenses	Number of licenses assigned.
Your license key	The license key provided by TEC-IT.

If you got a license for more than one installation the license key can be saved to the *.TWI file as well. Thus you do not have to input the license information on each target computer (see 8.1.1).

8 Supplement

8.1 .TWI Configuration File

When you save your settings (menu *File* ► *Save)* a configuration file will be created. This file is named with the ".TWI" extension and it contains various settings from the *Configuration* menu. So you have the possibility to create configuration files for different applications (or for different communication interfaces) and load them at a later time.

8.1.1 Licensing using the Configuration File

Usually the license key is saved in the registry once it has been entered inside the license dialog. However, if you purchase a Site license you can install the software on multiple workstations within the companies site and therefore you would have to do the licensing on each of these workstations. For easier deployment the license key can also be coded into a .TWI-file (provided the same .TWI-file is used for each installation). So you do not have to enter the license data on every single workstation, but it will be read from the .TWI Configuration file on *File* ► *Open.*

Example:

```
[License]
Licensee=My Company Inc.
LicKey=00000000000000000000000000000000
LicNo=1
KindOfLicense=Site
Product=TWedge Standard Edition
```

Licensee:	Name of the license holder (case-sensitive!)
LicKey:	License key
LicNo:	Number of licenses
KindOfLicense:	Type of license: Site or Single
Product:	Product name (optional)

Please contact our support or sales team if you want to use the product on multiple sites!

8.2 Output File

Buffered data is moved to the output file once the designated time interval has elapsed. The path, the file name and the extension can be set in the *Output File* Dialog. To automatically generate different file names you can use placeholders. These placeholders are associated with different values with will then be inserted into the output file name. (For a list of possible placeholders please refer to chapter 6.4).

Example:

If the following *File Name* is entered in the *Output File* dialog

```
logfile%mm%ss
```

…and the output file is created at 15:05:25, then the output file will be named "logfile0525".

8.3 Data buffer

The data buffer temporarily stores all data, which was read within a specified time interval. Once the time interval has elapsed, the data buffer is moved to the specified output file.

8.4 Function reference for pre-evaluation expressions

The following functions and constants may be used in pre-evaluation expressions:

8.4.1 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 $e^{\text{"Number"}}$, where e is the base of the natural Logarithms.
double	Exp10 («Number»)	Returns the $10^{\text{"Number"}}$.
long	Find («Text», «SearchText», «nStart»)	Seaches 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.
bool	IsLeapYear («Date»)	Returns TRUE if the specified date occurs within a leap year.
long	IsOdd («Number»)	Returns TRUE if "Number" is uneven.
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», «nStart», «nLenght»)	Returns the substring of string "Text" starting at position "nStart" with length "nLength". The first character of a string is located at postion 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 2: Supported Funtions for pre-evaluation expressions

8.4.2 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 3: Supported Constants for pre-evaluation expressions

9 FAQ

9.1 How can I stop the CR/LF after each data input?

Q: *I am attempting to route scanned data into an existing application, so I just need the raw data, no "Carriage Return" or "Line Feed". How can I turn the CR/LF off?*

A: Follow these steps to clip CR/LF at the end of your data:

- Open the menu *Configuration* ► *Interface* and change to the *Transmission* tab.
- Enter as for the Delimiter "\x0d\x0a" (these are the hex codes for CR+LF) (or just "\x0d" if you only need the CR stripped off).
- Leave the *Include Delimiter* option unchecked.

Now if a CR/LF is found in the input data it will be filtered out (because *Include Delimiter* is unchecked).

9.2 How can I select a certain input field automatically?

Q: *I would like to make sure that my data is always inserted into the same input field – in other words the cursor should always be placed in the correct field before the data is inserted.*

A: You can do this by setting a *Prefix* and a *Control Sequence.* (Please note that your application must allow you to place the cursor in the designated input field using a shortcut!)

- Open *Configuration* ► *Interface* in the menu and select the *Transmission* tab.
- Enter any character (or character string) of your choice in the filed *Data Prefix* – e.g."!1", then click OK. This character string is used as an internal placeholder and <u>should not be found in the actual data!</u>
- Open *Configuration* ► *Key Wedge Settings* in the menu.
- Select *Enable Control Sequences* in the *Control Sequence Table.*
- For the *Trigger Sequence* enter your selected Prefix (e.g. "!1").
- As *Action* enter the selected Shortcut designating the input field – e.g. {Alt-I}
- To accept your entry click on *Accept* in the *Control Sequence Table* and confirm with OK.

The selected Shortcut that designates the input field will now be inserted before each data package.

> ➢ To bring a window to foreground in combination with selecting an input field you can use the following Action command: {Act:MyWindowCaption}{Alt-i}

9.3 How can I bring a certain window to the foreground?

Q: *I want to make sure the data is entered into the correct window (= the correct application) each time, even if the window is deactivated or minimized.*

A: You can achieve this behaviour by setting up a prefix with a control sequence. The approach is similar to selecting an input field (see 9.2).

- First you have to learn the caption of the target window. Therefore press Alt+Tab (keep Alt pressed) and use the Tab key to switch to the desired window.

- Then go to the *Key Wedge Settings* dialog and insert a new Control Sequence. Use the “Act:” command (see 6.3.2.4) in combination with the window caption. It should look like one of the following samples: {Act:Word} or {Act:SAP R/3}
- Finally the Trigger Sequence must be specified as *Data Prefix.*
- You can also combine the Act-Command with the selection of an input field:
  e.g. {Act:MyWindowName},{ALT+i}
  The comma is used to insert a delay (see 6.3). Thus the Computer has enough time to execute the command.

9.4 Can I use two instances of TWedge simultanously?

Q: *I want to monitor two ports on my PC at the same time. Can I therefore run two instances of* TWedge *simultanously?*

A: Basically you can run two (ore more) instances of *TWedge* on one PC at the same time. Anyway there is one thing you should consider: Each instances must be configured to use a different interface. Otherwise the programs will block one another. (An interface can be used by one program at the same time only!).

> ➢ Note: Two instances cannot use the same hot-key shortcut!

10 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