

# Kofax Communication Server

## TC/FILBRK Technical Manual

Version: 10.3.0

Date: 2019-12-13

The KOFAX logo is rendered in a bold, blue, sans-serif typeface. The letters are thick and closely spaced, with a clean, modern aesthetic. The 'K' and 'F' are particularly prominent due to their size and weight.

© 2019 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

# Table of Contents

<b>Chapter 1: Preface</b> .....	<b>4</b>
Unicode Support.....	4
<b>Chapter 2: Configuration</b> .....	<b>5</b>
Installation with Novel Networks.....	5
<b>Chapter 3: Command Line Parameters</b> .....	<b>7</b>
<b>Chapter 4: Windows Code Page Support</b> .....	<b>8</b>
<b>Chapter 5: Functional Description</b> .....	<b>10</b>
Interaction with UAS User Module.....	10
Outgoing Messages.....	10
General Structure of Outgoing Message.....	10
Command Lines for TCFILBRK Process.....	11
Sending of LAN Break Messages ('B' Command Line).....	11
Copy Command ('C' Command Line).....	12
Windows Command Execution ('D' and 'E' Command Lines).....	13
Incoming Messages.....	13
Specifying API in Directories.....	13
API File Format.....	13
API files for Delivery Notifications.....	14
<b>Chapter 6: Errors During TCFILBRK Operation</b> .....	<b>16</b>
<b>Chapter 7: Registry</b> .....	<b>17</b>

## Chapter 1

# Preface

The TCFILBRK process is a native Windows application and belongs to the group of TCOSS processes. It can run either on the TCOSS Server (local) or on an external workstation (remote).

In both cases it is a client application connected to the main TCOSS server application via the TC/Transport Interface (TCTI).

TCFILBRK process performs these tasks:

- Sending of LAN break messages.
- API directory read – API in (routing of the messages from the LAN-application programs to the TCOSS).
- API directory write – API out (routing of TCOSS messages to LAN-applications).

TCFILBRK client communicates with the TCOSS server via the TCTI interface. On the other hand, the access to the network file server(s) is provided by the means of Windows network support.

## Unicode Support

Unicode is not supported.

## Chapter 2

# Configuration

TCFILBRK client communicates with the TCOSS server via the TCTI interface. On the other hand, the access to the network file server(s) is provided by the means of Windows network support.

On the TCOSS server, the UAS module must be configured for TCTI connection - UAS module on the Master without HW-SW assignment. Then the standard configuration for the TCFR should be chosen

With both working modes, local and remote, TCFILBRK process awaits its TOPCALL1.INI file in the Windows root directory. The only entry 'Path' in the section [TCFILBRK] is being used, which specifies the path to the TCOSS server.

Even if running local, the Path must be set to a valid TCOSS path like following:

```
[TCFILBRK]
Path=TCP/IP,193.81.166.121
```

The underlying TCTI interface (TCTI32.DLL) needs at least one entry 'LinkTypes' in the section [TCTI] in order to specify link types for communication with the TCOSS server.

Support operating systems:

- Windows Server 2008
- Windows Server 2012

## Installation with Novel Networks

For installations with Novel 4.1 server environments, you have to use IntraNetWare 4.11a or later on the workstation where TCLANPRT is running.

TCFILBRK may be started by TCSRVR, and even no autologin for NetWare and NT is required. The only requirement is that TCFILBRK processes must be started with a user account that is defined with the same user id and password under NT's user manager and also on Novel server. This NT user must have the right "logon as a batch job".

Please check if TCFILBRK process has following values of type string in its registry key configured:

- TCFILBRK\LogonType – set to "Batch"
- TCFILBRK\UserId – User Id of the NT and Novell 4 user
- TCFILBRK>Password – Password of the user (the same for NT and Novell 4!)

Of course, TCFILBRK should always use UNC paths to access network (e.g. Novell) resources, no logical drives.

**Note** Windows CSNW or GWSN must not be used, since they do not completely support NDS.

## Chapter 3

# Command Line Parameters

The TCFILBRK process can be started manually from the command lines or with TCSR service.

In both cases the following command line parameters are possible.

-gxx	channel group Channel group on the KCS to be queried via UAS module for incoming messages, for example -gA1 for channel group A1:. If not specified, the UAS TCOSS channel is being polled.
-epath	API out directory(ies). One and more API in directories can be specified as complete paths separated by comma (without blank), for example -eC:\api1,\\SERVER\vol1\api2. If not specified, no API in directory is being scanned. For network drives the UNC naming convention (e.g. \\TOPCALL\VOL2\API) must be used!
-u[DE]	use NT's command interpreter for execution of D or E commands (default: -uDE).

## Chapter 4

# Windows Code Page Support

All internal string manipulation routines support the current Windows System code page. This means e.g. that Japanese break messages are supported only if TCFILBRK is running on Japanese Windows.

In order to support multibyte characters (e.g. required for Japanese code page 932) the input and output code conversion tables of the asynchronous module (UAS) used by TCFILBRK must be set to transparent.

For standard installations (US Windows), code page 1252 is used. When using UAS module on KCS server with “TCFR Standard Configuration” for TCFILBRK (the only appropriate configuration supported by the Config program for TCFILBRK so far) all KCS text blocks (e.g. TCOSS 0 or TCOSS 1) are interpreted with code page 1252, which is not correct with special characters like äöüÄÖÜß, áé, etc.

In order to interpret also these special characters properly the output and input conversion tables within UAS module used for TCFILBRK must be configured manually. Without this configuration change, only all characters within the standard 7 bit ASCII character set are always correct (codes between 20h and 7Fh) since they are identical in all code pages.

### Required configuration change for TCOSS code page 0 and Windows code page 1252

UAS module outgoing conversion table, config lines 84 – 99:

```
:00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F ,084
:10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F ,085
:20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F ,086
:30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F ,087
:40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F ,088
:50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F ,089
:60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F ,090
:70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E D7 ,091
:C7 FC E9 E2 E4 E0 E5 E7 EA EB E8 EF EE EC C4 C5 ,092
:C9 E6 C6 F4 F6 F2 FB F9 FF D6 DC A2 A3 A5 50 83 ,093
:E1 ED F3 FA F1 D1 AA BA BF 5F AC BD BC A1 AB BB ,094
:5F 80 B4 70 6F AF AF B7 A8 A8 B0 B8 B7 22 B8 AF ,095
:70 A7 D0 69 48 D7 55 4C 4C B3 8C 6B DE 54 6E 6E ,096
:6B A4 F0 F0 68 69 49 49 49 F8 9C BE FE 74 6E 41 ,097
:42 DF 47 44 45 5A B5 45 54 49 4F 4B 4C D8 4D 4E ,098
:58 B1 4F 50 52 53 F7 54 B0 59 46 48 50 B2 6C 20 ,099
```

UAS module incoming code conversion table, config lines 105 – 120:

```
:FF FF FF FF FF FF FF FF 0F FF FF FF FF 12 FF FF ,105
:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ,106
:20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F ,107
:30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F ,108
:40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F ,109
:50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F ,110
:60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F ,111
:70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 5F ,112
:B1 5F 2C 9F 22 5F 5F 5F 5E 5F 53 3C CA 5F 5F 5F ,113
```



```
:5F 60 B2 22 22 BC 2D 2D 7E 5F 73 3E DA 5F 5F 59 ,114
:20 AD 9B 9C D1 9D 7C C1 B8 63 A6 AE AA 2D 52 B5 ,115
:BA F1 FD C9 B2 E6 5F BC BB 31 A7 AF AC AB DB A8 ,116
:41 41 41 41 8E 8F 92 80 45 90 45 45 8D A1 8C 8B ,117
:C2 A5 4F 4F 4F 4F 99 C5 ED 55 55 55 9A 59 CC E1 ,118
:85 A0 83 61 84 86 91 87 8A 82 88 89 8D A1 8C 8B ,119
:D3 A4 95 A2 93 6F 94 F6 D9 97 A3 96 81 79 DC 98 ,120
```

**Required configuration change for TCOSS code page 1 and Windows code page 1250**

UAS module outgoing conversion table, config lines 84 – 99:

```
:00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F ,084
:10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F ,085
:20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F ,086
:30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F ,087
:40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F ,088
:50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F ,089
:60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F ,090
:70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E D7 ,091
:C7 FC E9 E2 E4 F9 E6 E7 B3 EB D5 F5 EE 8F C4 C6 ,092
:C9 C5 E5 F4 F6 BC BE 8C 9C D6 DC 8D 9D A3 D3 E8 ,093
:E1 ED F3 FA A5 B9 8E 9E CA EA 20 9F C8 BA D4 D1 ,094
:5F 80 B4 83 88 C1 C2 CC AA 20 F1 F2 8A AF BF 9A ,095
:98 49 20 69 C0 DA C3 E3 E0 DB FD 6B DD FE A7 FB ,096
:F0 D0 CF CB EF D2 CD CE EC 20 D8 F8 67 DE D9 41 ,097
:42 DF 5F 5F 45 5A B5 48 5F 49 5F 4B 5F 47 4D 4E ,098
:5F 27 4F 5F 50 5F F7 54 B0 59 5F 58 5F 20 91 20 ,099
```

UAS module incoming code conversion table, config lines 105 – 120:

```
:FF FF FF FF FF FF FF FF 0F FF FF FF FF 12 FF FF ,105
:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ,106
:20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F ,107
:30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F ,108
:40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F ,109
:50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F ,110
:60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F ,111
:70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F ,112
:B1 5F 2C B3 22 5F 5F 5F B4 5F BC 3C 97 9B A6 8D ,113
:FE 60 B2 22 22 2E 2D 2D C0 5F BF 3E 98 9C A7 AB ,114
:20 5E 5E 9D 5F A4 7C CE 20 43 B8 3C 2D 2D 52 BD ,115
:F8 20 20 88 B2 E6 5F 2E 20 A5 AD 3E 95 22 96 BE ,116
:C4 B5 B6 C6 8E 91 8F 80 AC 90 A8 D3 B7 D6 D7 D2 ,117
:D1 AF D5 9E AE 8A 99 2A DA DE C5 C9 9A CC DD E1 ,118
:C8 A0 83 C7 84 92 86 87 9F 82 A9 89 D8 A1 8C D4 ,119
:D0 BA BB A2 93 8B 94 F6 DB 85 A3 CF 81 CA CD 20 ,120
```

## Chapter 5

# Functional Description

This section describes the functional description.

## Interaction with UAS User Module

During startup, TCFILBRK process connects via transport interface TCTI to one of the UAS-TCTI modules on the KCS server side. UAS-TCTI are those asynchronous modules, which were configured for TCTI by the WCONFIG program.

Basically used for transferring **pure text and TCI image messages** from the KCS to the LAN file server(s) and vice versa. Within the TCFILBRK, **no code/image conversion is being performed**, the only possibility to perform some conversions in the messages are inbound/outbound conversion tables in the UAS module. Even the TCI image files, transferred via this link, are handled in the same way as all other text messages by the TCFILBRK.

## Outgoing Messages

This section describes about outgoing messages.

### General Structure of Outgoing Message

The outgoing messages transferred via UAS to the TCFILBRK process must have the following structure:

```
[Dcommand line]
...
[Dcommand line]
[Ecommand line]
...
[Ecommand line]
B/Ccommand line
anyline 1
...
anyline z
```

**D-**, **E-**, **B-** and **C** command lines are message specific command lines (those in [] brackets are optional). **anyline 1-z** are all possible text, control (++) and TCI code lines. **txtcntrl 1-y** are text control lines.

**Following general rules on setting up an incoming message apply:**

- Each message must start immediately with one of the message command lines.
- Each message command line must start on the very first position in the line.
- Each message must contain just one of the 'C' or 'B' command lines.

- The 'C' or 'B' command line in the message can be (optionally) preceded by up to 5 'D' and /or up to 5 'E' command lines.

## Command Lines for TCFILBRK Process

There are following message commands defined:

### Main action command lines

- **B** command - Send a LAN break message.
- **C** command - Copy to the API-out directory

### Support command lines

- **D** command – Windows internal/external command execution before the main action specified via P/C line has been started.
- **E** command - Windows internal/external command execution after the main action specified via P/C line has been done.

Commands to be executed include internal commands such as dir, .COM, .EXE programs or even .BAT file. The manner how this command/program file is executed depends on the TCFILBRK command line parameter -u[DE].

There are two possibilities:

- If the -u was specified for specific command line(s) (e.g. -uD for the D command line), the command of this line is executed by passing it to the CMD.COM Windows command interpreter. Thus, **all kinds of commands (Windows internal/external, .COM, .EXE and .BAT files)** can be executed, but **no** return code of the command/program can be checked. This means, that the **send order on the KCS does not terminate with an error code, if the executed command/program terminates with an error.**
- If -u switch was not specified for specific command line(s) (e.g. -uE for D command line) the **command of this line is executed directly by starting it as a thread process under Windows.** Thus, only external command, .COM and .EXE program files can be executed. Return code of the command/program is checked, so that the send order on the KCS terminates with corresponding error code (for specific error codes refer to description of the command line).

## Sending of LAN Break Messages ('B' Command Line)

The B command line in the incoming message invokes sending of a LAN break message.

Starting on the second position in the line, it should contain the corresponding command for sending a break message.

Novell Netware:	Bsend "message" to LANUserID	
Lan Manager:	Bnet send LANUserID "message"	Windows Server 2003 R2 and earlier
Lan Manager:	Bmsg LANUserID "message"	Windows Server 2003 and later

The command behind the 'B' is executed by the TCFILBRK.

### Substitution within the Break message

If the command part of the 'B' command line contains one or more '%n' sequence(s), where  $0 < n < 10$ , the '%n' sequence will be replaced by the n-th text line in the message counted from the end of the B command line.

### Example1 (with %n substitution)

An appropriate mask (e.g. +B) for a Novell network could look like this:

```
BSEND "%3" TO $N1243$
$X$
```

the send order is then :

```
..S,R=test,N=06:+BJM
```

The third line of document TEST will be displayed on the screen of user JM.

### Example 2 (without substitution)

A appropriate mask (e.g. +B) for a Novell network could look like this:

```
BSEND "$B$" TO $N1243$
```

the send order is then

```
..S,R=test,N=06:+BJM(Msg. received)
```

Message "Msg. received" will be displayed on the screen of user JM.

**Note** Above examples assume that channel 06: is polled by the TCFILBRK.

## Copy Command ('C' Command Line)

Generally, the Copy command has the following syntax:

```
Cpath/outfile
```

The first string behind the 'C' specifies the complete path to the output file (incl. file name 'outfile'), is mandatory. **Consider that in the path specification, as to delimit different directories, the slash ('/') is used, not backslash ('\).** For network drives the UNC naming convention (e.g. \\TOPCALLVOL2\API\File.1) must be used!

### Generation of the output file name

The 'outfile' file name parameter could be generated for example by means of the KCS mask parameter \$R\$. As the length of the KCS's message name exceeds the length of the DOS file names (max. 8 characters), following method to convert the 'outfile' parameter into the DOS file name is used in the TCFILBRK:

If the 'outfile' parameter has the form 'xyz.ext', the DOS output file will be 'xyz.ext'. For example, such a C command line could be generated by the KCS mask line

```
Cc:/apidir/$R0209$.TIF
```

If the 'outfile' parameter has the form 'xyz', does not contain the '.' character at all and is shorter than 8 characters, the DOS output file name will be 'xyz'.

If the 'outfile' has the form 'xyz', does not contain '.' character and is longer or equal to 8 characters, the '.' character is inserted to the 3rd position from the end of 'xyz' string. For example, for the 'outfile' parameter 'abcdefghijkl' the DOS output file will be 'abcdefgh.ijk'.

Please note that blanks within the filename are not removed.

### Copy function of the 'C' command

All message lines starting with the first line behind the 'C' command line are transparently copied to the output file without any image conversion.

## Windows Command Execution ('D' and 'E' Command Lines)

The command starting on the 2-nd position just behind the 'D' or 'E' letter is executed by the TCFILBRK, those starting with 'D' before the main action is started, those with 'E' after the main action has been **successfully** completed.

It should be taken into account, that once the 'D' command has been executed, another 'D' , 'P' , 'B' or 'C' command line is expected. If the line with an invalid command identifier follows the last D-command line in the message, the send order on the TCROSS terminates with error code R1 - illegal 1st line.

If 'D' is not specified within the -u TCFILBRK command line parameter and invoked command/program returns a non zero error code an **application error** (R9) is reported. If the command could not be executed (e.g.: because off too less memory) **can't execute** (R8) will be reported. (The break-code off both errors is 2.)

## Incoming Messages

This section describes about incoming messages.

### Specifying API in Directories

When TCFILBRK is started with command line parameter -e, for example:

```
TCFILBRK-ec:\api1[,\\topcall\vol2\api2[,d:\api3]]
```

the program scans specified directories, which are delimited by comma (without blank). If it finds a file in one of these directories, the text and send orders of this file are transferred to the KCS.

Such files are referred to as API files.

### API File Format

The API file consists of one or more text blocks and associated send order(s). The format is:

```
[  
  Sendorder  
  [Sendorder..]  
  Text  
  [Text..]  
  ..]  
]
```

Where each text will be sent to all associated send orders.

The format of the send order is

```
DN=number,U=userid,D=date,T=time,TERM=term
```

or

```
SN=shortnumber
```

or

```
SN=shortnumber.userid
```

The above lines must start at the very beginning of the line. If the user is not specified, default user is APIOPER. If there are no send orders specified, the file is sent to the APIOPER user by default (with one exception - API files for delivery notification).

**Examples:**

```
DN=F:6613321
DN=FAX$6613321
DN=FAX$6613321,U=SL
SN=SHORT (the same as DN=.SHORT - search in NN99)
SN=SL.+TCTECH (address book entry in system address book)
SN=SL.USER5 (address book entry in personal address book of USER5)
```

## API files for Delivery Notifications

Files with extension .NFC in the API-read directory (API-in) are handled as files for delivery notification.

The file for delivery notification must have one of the following formats (for more detailed description, refer to the *TCOSS System Manual*).

For delivery notification

```
1.2: TCOS      DELIVERY NOTIFICATION
1: CORRELATION INFORMATION:
   92-01-09-14:16/0127  date & time of forwarding/serial #
3: DELIVERY INFORMATION:
   = 135361 tpcal a
4: TIME OF DELIVERY: 92-01-09-14:19
6: NOTE: 0 08
```

For non delivery notification

```
1.3: TCOS      NON DELIVERY NOTIFICATION
1: CORRELATION INFORMATION:
   92-01-09-14:16/0127  date & time of forwarding/serial #
3: DELIVERY INFORMATION:
   = 135361 tpcal a
4: TIME OF DELIVERY:
6: NOTE: 0 08
```

If any file with the extension .NFC is found in one of the API-in directories, a special notification logon is done (..LOGON, TYP=3) as to force the KCS server to evaluate this notification, match it to an open entry of status 'at next node'. When a matching entry is found it is terminated positively or negatively according to the delivery or non delivery notification.

With delivery notification files no send orders are generated, even though specified in the beginning of the file. Also if no send orders are specified in the beginning of the file, no send order to the APIOPER user is generated.

**Note** The only criterion for TCFILBRK whether the specific file is a notification or not, is its extension. No checks for the contents of the notification are performed in the TCFILBRK program.

Basically, this feature is used for terminating send orders on the KCS which were generated for KCS documents with the Convert command extended with optional parameter -f (TCLANPRT process).

Such send orders should use special mask to generate not only the Convert command line, but to provide the application program with all necessary correlation information as well. This correlation information will be then used by the application to generate the proper notification file.

**Example:**

The special mask (+4) on the KCS could look like:

```
CJ:/SB/TCFILBRK/SS$.DAT -f
CORRELATION INFORMATION: $I0102$-$I0304$-$I0506$-$I0708$: $I0910$/ $I1318$
$X$
```

Assume, the document named TEST with the only text line 'Test' is being sent with the following send order, and the serial number of the send order will be (for example) 8337:

```
..S,R=TEST, N=A1:+4
```

(TCOS channel A1: is the channel being queried via UAS by the TCFILBRK).

Thus, due to the -f switch within the +4 mask, the send order will go to 'at next node' state.

The file 8337.DAT will be generated in the directory J:\SB\TCFILBRK, the contents of this file looks like:

```
CORRELATION INFORMATION: 95-01-18-18:40/8337
Test
```

The application program in the network should analyze this correlation information provided in the file 8337.DAT and use it to create corresponding notification file (with extension .NFC), such as

```
1.2: TCOS          DELIVERY NOTIFICATION
1: CORRELATION INFORMATION:
   95-01-18-18:40/8337    date & time of forwarding/serial #
3: DELIVERY INFORMATION:
   = 135361 tpcal a
4: TIME OF DELIVERY: 95-01-18-18:42
6: NOTE: 0    08
```

and put in into API-in directory.

After this file has been sent to the KCS, the original send order which generated the 8337.DAT file on the file server, goes to positively terminated.

## Chapter 6

# Errors During TCFILBRK Operation

During the TCFILBRK's operation, following error situations can occur:

### **JOB (printing, copying or converting) specific errors.**

If such an error occur, the corresponding send order on the KCS will end with one of the following error codes. With all error codes except the 'user break' TCFILBRK itself recovers and goes on further.

Answerback field	Error	Break	Meaning
illegal 1st line	R1	5	document empty or first line does not start with C or P
can't capture	R2	4	capture command is not successful
can't print	R3	3	error during printing
file already exists	R4	2	DOS file (specified with C command line) already exists
no write permission	R5	4	insufficient rights or directory does not exist
break during copy	R6	2	error during copying (perhaps disk full)
break while printing	R7	2	break while printing
can't execute	R8	2	Can't execute DOS command
application error	R9	2	application error (e.g. wrong command line)
unknown image format	RA	5	image format not supported (C command, -d switch)
too many commands	RB	5	too many commands lines
output file error	RC	1	output DOS file error (during converting)
temp. file error	RD	1	temporary DOS file error
user break	RE	1	TCFILBRK stopped by user during printing
time-out	RF	1	time-out in TCFILBRK – protocol problem
TCTI attach error	RG	1	can't attach to TCTI
TCTI disconnect	RH	1	TCTI disconnect error
internal error 1	RI	1	TCFILBRK internal error (wrong event)
internal error 2	RJ	1	TCFILBRK internal error (wrong state)
ULL time-out	RK	1	ULL time-out protocol problem
ULL TCTI disconnected	RL	1	TCTI disconnect occurred during printing
ULL protocol error	RM	1	ULL protocol error



## Chapter 7

# Registry

All registry settings specific to TCFILBRK are located in sub-key:

```
HKEY_LOCAL_MACHINE\Software\TOPCALL\{ApplKey}
```

{ApplKey} is the Root application key (usually TCFILBRK).

(In previous releases before 7.10.00, TCFILBRK used the Topcall1.ini file to store some settings. With the new release, all parameters are stored in the registry).

Parameter	Registry value	Type	Used for
SetupAPIInDir	[ApplKey]\SetupAPIInDir	String	File API directory
SetupChannelGroup	[ApplKey]\SetupChannelGroup	String	channel polled by TCFILBRK
Transport	{ApplKey}\TCTI\Transport	String	TCTI transport (native or PRC)
Linktypes	{ApplKey}\TCTI\Linktypes	String	TCTI link types
Path	{ApplKey}\Topcall\Path	String	Path to TCOSS

Refer to *TCOSS Configuration Manual* and *TCSR V Manual*(Kofax Communication Server Service User Manual) for further registry values.