

Kofax Communication Server

TCOSS Message Preprocessor 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

Chapter 1: Preface	5
Chapter 2: Installation	6
Prerequisites.....	6
Hardware (Digital Signature Only).....	6
Software.....	6
Prepare Hardware (Digital Signature Only).....	6
digiSeal Server (Digital Signature Only).....	7
TCOSS.....	7
Service Configuration.....	7
TDCDSIGN.....	8
Digital Signature Parameters (Digital Signature Only).....	9
Windows Authentication Parameters.....	10
Post Installation Steps (Digital Signature Only).....	10
Chapter 3: Digital Signature	11
Legal Background.....	11
digiSeal Server.....	11
Overview of Operation.....	11
Signed Invoice Example.....	12
General Functionality.....	12
Requesting a Signature for a Message.....	12
File Interface to digiSeal Server.....	14
Resulting Message.....	16
Error Handling.....	17
Optional Asynchronous Run Mode.....	17
Fax Signature.....	18
Message Format Considerations.....	18
Configuring the Layout.....	18
Image Resolution.....	23
Technical Limitations Concerning Cover Sheets and Overlays.....	24
Adjusting Invoice Content (If Necessary).....	24
File Interface to digiSeal Server.....	25
Resulting Message.....	25
Verifying the Received Fax.....	25
Email Signature (PKCS#7).....	26

Signature Mode.....	26
File Interface to digiSeal Server.....	27
Resulting Message.....	27
Email Signature (PDF with 2D Barcode).....	28
Signature Mode.....	29
Resulting Message.....	30
Email Signature (Adobe PDF).....	30
Signature Mode.....	31
Resulting Message.....	31
Chapter 4: XML Attachment Parsing.....	32
Overview of Operation.....	32
XML Exit Configuration.....	32
Schema Validation.....	34
Chapter 5: Maintenance.....	35
Registry Keys.....	35
Event Log Entries.....	38
Trace File.....	38

Chapter 1

Preface

TCOSS optionally routes send orders to a special queue, where they can be examined and changed before delivery to the target queue.

The decision which messages shall be routed in this way and how they are processed is defined by the capabilities of the recipient service.

The TCOSS Message Preprocessor reads the queue and changes the messages in an appropriate way.

Currently, two modes of message processing are supported:

- Adding a digital signature to the message
- Parsing an XML attachment

Both actions can be applied to the same message.

Digital signature:

The following signature types can be added to a message:

- Graphical fax signature
- PKCS7 email signature (TIF or PDF message image, or individual attachments)
- PDF file with graphical 2D signature for laser printer
- Adobe PDF version 1.6 signature

XML Parsing:

The first XML attachment in a message can be examined prior to sending. Information about the XML content is written to the message header. Optionally, the XML file is validated against a schema.

Important The Kofax Communication Server and its components formerly used the name TOPCALL. Some screen shots and texts in this manual may still use the former name.

Chapter 2

Installation

Use this chapter to install TCOSS Message Preprocessor. If you don't need the feature Digital Signature, you can skip the sections designated as "Digital Signature Only".

Prerequisites

This section describes the prerequisites.

Hardware (Digital Signature Only)

Chip card for server-based creation of qualified signatures (includes certificate): available from a trust center like D-Trust

Chip card reader

Software

XML parsing:

- All KCS components must be updated to version KCS 10.0.0 or later.

Digital signing:

- digiSeal Server version 1.06.01 or later
- TCOSS version 7.59.06 or later
- TCLINK version 2.14.03 or later
(if using ++DIGISIGN feature)

Prepare Hardware (Digital Signature Only)

You need a computer for digiSeal server. Refer to digiSeal Server manual for hardware and software requirements, and also for security requirements. DigiSeal Server usually runs on a dedicated machine (not on the TCDCSIGN computer).

For each customer:

- Purchase a chip card for (qualified) mass-signatures. (German laws require a 'qualified' signature. In other countries, legislation might be different, e.g. requiring only an 'advanced' signature).
- Install a chip card reader (usually with USB or serial interface) on the digiSeal Server computer.

Note For multiple customers, it is recommended to use the “Secrypt Signature Card Unit”, – a USB device that combines up to 10 individual chip card readers.

This device is available at secrypt.

Please consider that the delivery time for chip cards and for the Secrypt Signature Card Unit can be 2 to 3 weeks.

digiSeal Server supports chip card readers of class 1 (without keypad, without display), class 2 (with keypad, without display) and class 3 (with keypad and display). The supported communication standards for the chip card readers are PC/SC and CT-API, via USB or serial port.

digiSeal Server (Digital Signature Only)

Install digiSeal Server according to its documentation. Configure the software (assign chip card readers to customers).

TCOSS

Configure SYSCONF line 17, document converter level. Possible values:

- 00: both digital signature and XML parsing disabled
- 04: digital signature enabled
- 08: XML attachment parsing enabled
- 0C: both digital signature and XML parsing enabled

For digital signing you need the license TC Electronic Fax Signature. Licensing is done per workstation. In an ASP environment, you need one license for each TCOSS instance.

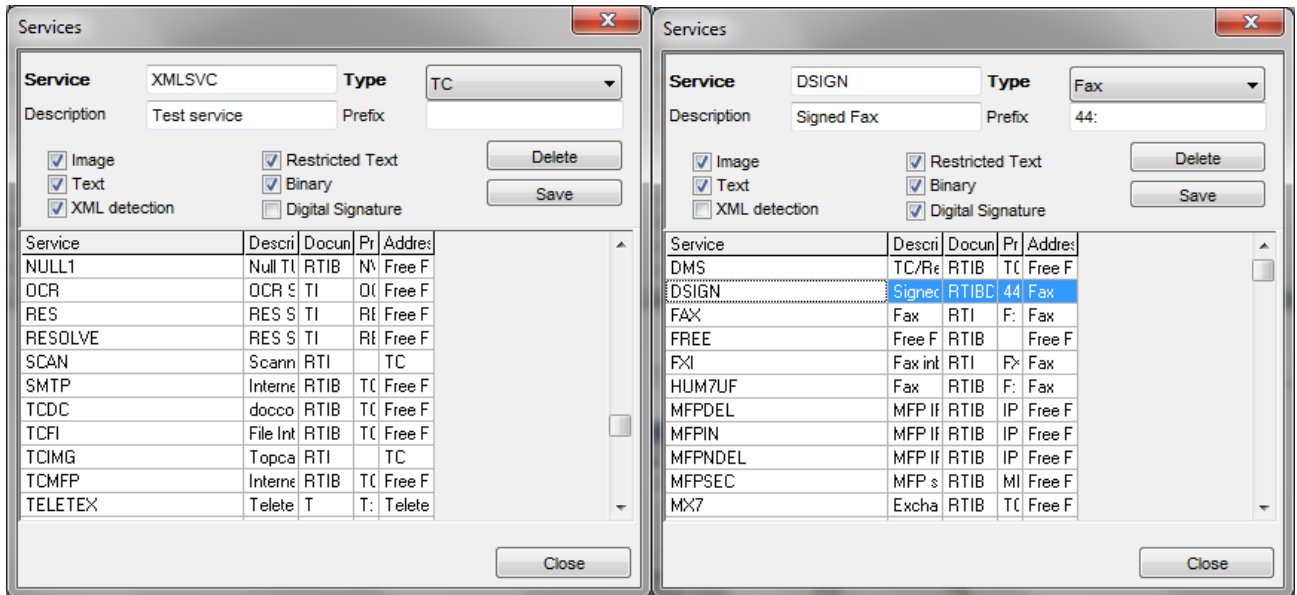
The XML parsing feature requires no separate license.

Service Configuration

Use TCfW to configure a service for XML and/or digital signature.

- If you select “XML detection” for a recipient service, messages are routed to the TCOSS Message Preprocessor for XML parsing.
- If you select “Digital Signature” for a recipient service, messages are routed to the TCOSS Message Preprocessor for digital signing.

Examples:

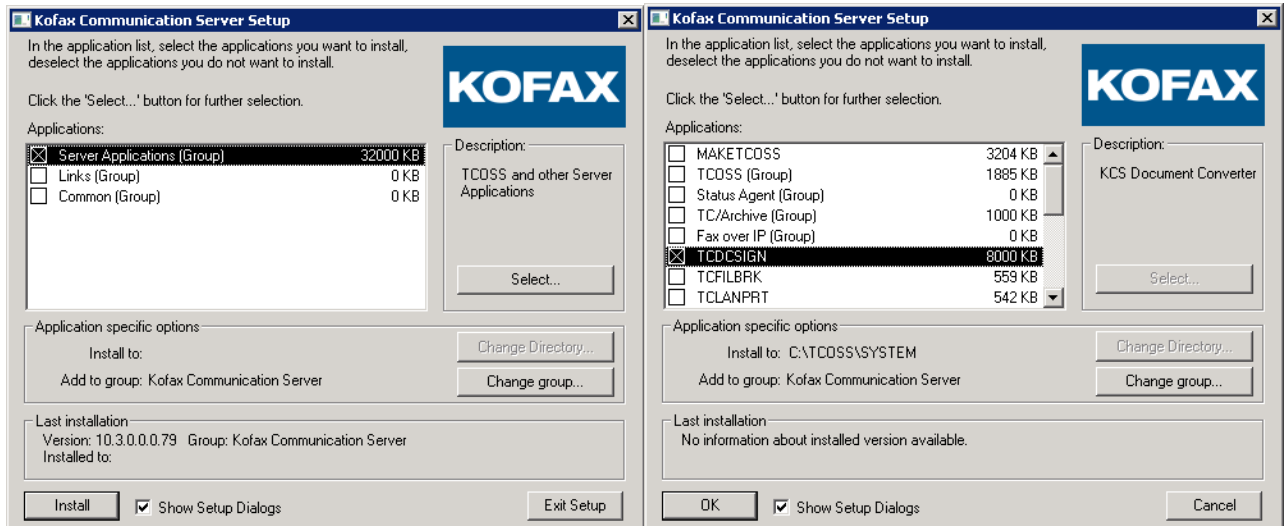


In an ASP environment, create a service for each TCOSS instance.

TCDCSIGN

For each TCOSS instance, you need one or more (for load sharing) TCDCSIGN instance(s).

Start KCS setup, and select group Server Applications (Group). Click **Select**. From the server applications, select **TCDCSIGN**.



Click **OK**.

If you want to use the Digital Signature feature, make sure that **Configure for Digital Signature** is selected. If you only want to use XML parsing, you can clear the checkbox.

Setup asks for the KCS account used by TCDCSIGN.

Click **OK**, then click **Install**.

Digital Signature Parameters (Digital Signature Only)

Setup asks for the most important configuration settings for digital signatures:

Common part of interface folder path (Sign\RootDir):

This folder holds one subfolder per customer, the interface directories for digiSeal server are below the subfolders. These folders must be created manually.

Default customer name (Sign\DefaultCustomer):

If there is only 1 customer per TCOSS, or if the customer name of the sender cannot be found (e.g. because the sender has no KCS user profile), TCDCSIGN uses the interface directories of the default customer.

User profile field holding customer name (Sign\CustomerField):

Choose “Only 1 customer” if the TCOSS instance belongs to a single customer (is translated into an empty string in the registry value). Otherwise choose one of the other options, e.g. Company, Department, Salutation.

Timeout for signature creation (sec) (Sign\WaitTimeSec):

Maximum wait time for every signature creation.

Windows Authentication Parameters

Setup then asks for the Windows account used by TDCDSIGN. This account must have full access to the interface directories, logon as a batch job permissions and must be a member of the local Administrators group.

The screenshot shows a dialog box titled "TDCDSIGN - Windows user". Inside the dialog, the text "Enter or modify the parameters below" is displayed. There are three input fields: "Windows NT User Account", "Domain (blank for local computer)", and "Password (default or '*' leaves existing setting)". The password field contains an asterisk (*). At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Post Installation Steps (Digital Signature Only)

1. Start TDCDSIGN. It will create all mandatory registry values automatically.
2. Create the default interface directories, and optionally a default LAYOUT.INI file.

These directories must be accessible for digiSeal Server and for TDCDSIGN.

3. If the TCOSS instance holds more than 1 customer:
 - configure the field holding the customer name (*Sign\CustomerField*)
 - create a set of interface directories for each customer (base directory with IN, OUT and CFG subdirectories).
These directories must be accessible for digiSeal Server and for TDCDSIGN.
 - if the interface base directory name is different from the customer name: configure the mapping between directory and customer name in registry values Sign\Dir\ - optionally create a default layout file LAYOUT.INI in the interface base directory of every customer.

Chapter 3

Digital Signature

This section describes information about Digital Signature.

Legal Background

Since January 2004, German laws restrict sending invoices via fax servers:

When you receive an invoice via a fax server, reimbursement of the VAT (Value Added Tax) is only possible if the fax has a graphical signature that guarantees the identity of the sender and the integrity of invoice data.

To conform to the law, this graphical signature must be equivalent to a qualified digital signature, produced by a secure chip card.

Therefore, the first version of this product was used to put a signature on outgoing faxes.

In the meantime, TCDCSIGN also supports several types of email signature.

digiSeal Server

Company secript® is specialized on software solutions for signature and electronic invoicing. Their product digiSeal Server™ is used for mass-signing electronic messages.

digiSeal Server supports several types of email signatures, as well as a graphical signature (two-dimensional bar code, proprietary format), that is inserted into every page of the signed document.

The graphical signature is optimized for textual information. The bar code contains the certificate of the sender, the time of signing and a graphic image of the complete signed part of the document.

The recipient of the signed fax can use a free utility (digiSeal Reader) to verify the signature and to compare the received fax with the graphic image stored in the bar code.

KCS uses TCDCSIGN as an interface to digiSeal Server.

Overview of Operation

Messages submitted to TCOSS can be marked as needing a signature.

It is possible to specify the type of the signature, as well as the scope of the signature (e.g. whole message, single attachment).

TCOSS forwards such messages to an internal queue that is polled by the TCOSS Message Preprocessor.

The TCOSS Message Preprocessor has a file interface to the digiSeal signature server. TCDCSIGN has to provide the object to be signed in an input directory and then receives the signed object in an output directory. Directory names and file formats depend on the signature type.

For the fax signature, the process is more complicated. Several input files must be provided:

- TCDCSIGN creates a multi-page TIF file that contains the part of the message needing signature. One or more other TIF files are provided, containing writing paper information. Layout information (margins, names of writing paper files) is provided in an INI file.
- digiSeal server creates a result file (also multi page TIF). On every page of this result file, input data, writing paper information and a graphical signature are merged together. The input data is scaled to fit into the available space.
- TCDCSIGN then converts the TIF file to TCI code, packs the TCI code into an attachment and replaces the part of the message needing signature with this attachment.
- The original unsigned TIF file is added as an alternative representation of this attachment.

By default, TCDCSIGN waits for the signed file, with a configured timeout. If the timeout expires, the send order is terminated negatively.

Optionally, TCDCSIGN can work asynchronously, i.e. provide input file(s) to be signed, set the send order to "Retry" and proceed with the next send order. When the send order is retried, TCDCSIGN checks if the signed output file is ready.

Signed Invoice Example

General Functionality

This section described the general functionality of TCDCSIGN.

Requesting a Signature for a Message

Only certain messages are routed to TCDCSIGN for signing:

- Messages sent to a recipient service with "Digital Signature" capability
- Messages that contain a ++DIGISIGN directive and were posted by TC/LINK or UAS.

Note With TC/LINK-MX and TC/LINK-LN, ++ directives in RTF message body are not evaluated!

Service capability “Digital Signature”

You can use TCfW to set this service capability. For instance, you can create a service FAXSIGN for fax invoices that must be signed and use the service FAX for normal fax messages.

++DIGISIGN directive

In some cases, it is not possible to use a dedicated service for signed messages. For instance, TC/LINK-AC only knows one fax service.

A fax may also hold several invoices. Then every contained invoice must be signed separately.

In these cases, you would use the ++DIGISIGN directive. (Note: only with TC/LINK or UAS, not with TCFW, not with TC/LINK-MX or TC/LINK-LN RTF text)

Syntax for fax signature: ++*DIGISIGN* <layout-info>

See below for details about the layout-info parameter.

Syntax for email signature:

++DIGISIGN parameter	Signature type	Input	Output
@ATT	PKCS7 (per attachment)	Any file attachment	PK7 attachment
@MSG_TIF	PKCS7 signature for TIF image of the message	Part of the message (image view)	PK7 attachment
@MSG_PDF	PKCS7 signature of PDF image of the message	Part of the message (image view)	PK7 attachment
@MSG_2D	PDF with 2D barcode	Part of the message (image view)	PDF attachment
@ATT_2D	PDF with 2D barcode	TIF G4 attachment (resolution as high as possible)	PDF attachment
@ATT_PDF	Adobe PDF signature	PDF attachment	PDF attachment

See the individual sections below for details about the parameters.

Parameter scope:

A single message can contain several ++DIGISIGN commands. Every command is valid until the next such command is encountered. The signature will be applied to the message part that follows.

For messages submitted via TC/LINK, the ++DIGISIGN command can also be part of a text attachment (+ + option attachment, see TC/LINK manual).

Configuring a default signature mode for a service

You can configure TDCDSIGN to use a default signature mode for certain recipient services. Thus, it is not necessary to use a ++ command for the email signature types.

The service must have the “Digital Signature” capability.

If the message contains a ++DIGISIGN command, TDCDSIGN ignores the registry setting for the recipient service.

The mapping between recipient service and signature mode is done via optional registry values *Sign \Service_XXXX* of type REG_SZ:

Examples:

```
Service_SMTP01 = @ATT_PDF
Service_SMTP02 = @ATT
```

```
Service_SMTP03 = @ATT_2D
```

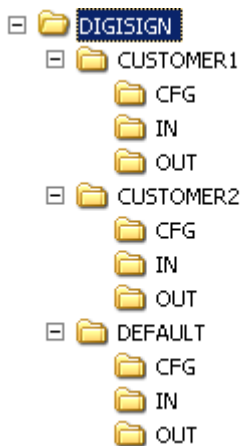
In the same way, you can specify the layout name for the fax signature:

```
Service_FAXSIGN = FS/LAYOUT1
```

The registry values can be created via a registry editor.

It is not necessary to restart TDCSIGN after adding or modifying these registry values.

File Interface to digiSeal Server



Basically, the file interface between digiSeal Server consists of a set of directories:

Signature type	Default directories digiSeal server 1.x	Default directories digiSeal server 2.x	Registry values for configuration
Fax	IN OUT CFG	IN OUT CFG	Sign\DirFaxIn Sign\DirFaxOut Sign\DirCfg
Email (PKCS#7)	BIN_IN BIN_OUT	IN OUT	Sign\DirBinIn Sign\DirBinOut
Email (PDF with 2D barcode)	2D_IN 2D_OUT	IN OUT	Sign\Dir2DIn Sign\Dir2DOut
Email (ADOBE PDF)	PDF_IN PDF_OUT	IN OUT	Sign\DirPdfIn Sign\DirPdfOut

Note In asynchronous mode (see section [Optional Asynchronous Run Mode](#)), the CFG directory is needed for the email signature types also.

- Files in the input directory are written by TCDCSIGN and removed by digiSeal Server.
- Files in the output directory are written by digiSeal Server and removed by TCDCSIGN.
- Files in the additional CFG directory are written and removed by TCDCSIGN.

Normally, only one set of these folders is necessary (folders below DIGISIGN\DEFAULT).

If the TCOSS instance polled by TCDCSIGN holds more than 1 customer (certificate owner), there must be a separate set of interface folders for every customer. (for instance CUSTOMER1, CUSTOMER2 in the above picture). A separate signature card and card reader is needed for every customer.

The TCDCSIGN registry configuration holds the mapping between directory names and customer names: The customer name must be stored in a text field in the originator's user profile (field configurable per TCDCSIGN instance via registry key *Sign\CustomerField*).

For messages from senders without a shadow user, TCDCSIGN uses the default customer's settings (*Sign\DefaultCustomer*).

You can define the name of the customer's interface folder (relative to a configurable root folder) in registry value *Sign\Dir\<CustomerName>*. If this value does not exist, TCDCSIGN uses the customer name as the folder name.

Examples:

Sign\CustomerField = "Company"

User *FS* sends a fax that must be signed.

The company field in the user profile of FS is "*Kofax*".

Registry value Sign\Dir\TOPCALL is "*Kofax*".

Registry value Sign\RootDir is "C:\DIGISIGN".

-> TCDCSIGN uses folders below "C:\DIGISIGN\Kofax" for signing this message.

Sign\CustomerField = "Company"

User *Miller* sends a fax that must be signed.

The company field in the user profile of user Miller is "*ABB*".

Registry value Sign\Dir\ABB does not exist, registry value Sign\RootDir is "C:\DIGISIGN".

-> TCDCSIGN uses folders below "C:\DIGISIGN\ABB" for signing this message.

Sign\CustomerField = "" (means that there is only 1 customer on this TCOSS)

Any user sends a fax that must be signed.

Registry value *Sign\DefaultCustomer* is "DEFAULT".

Registry value *Sign\RootDir* is "C:\DIGISIGN".

-> TCDCSIGN uses folders below "*C:\DIGISIGN\DEFAULT*" for signing this message.

Resulting Message

TCDCSIGN converts the signed file provided by digiSeal Server to KCS format and stores it in the message.

For graphical signatures and per-message email signatures, the original content of the area to be signed is removed, - only an image of the unsigned message part is stored as an alternative content.

For email signatures applied to individual attachments, the unsigned attachment is stored as an alternative content of the signed attachment.

For details see subsequent sections.

Note

The changes are stored in the document on TCOSS. This means that (except for attachment-based email signature) the original content is lost. If you want to keep the original document, you should send the message to a user who has an IN-action to the real recipient. Thus, only the content of the in-action is overwritten, not the original message in the sender's outbox.

TCSI Fields in Alternative Content Representations

The value of field *TS_APPL_ID* is always "TOPCALL Document Converter".

	TS_FILE_NAME	TS_CONTENT_TYPE
Signed image	TCOSS filename + extension TCI, e.g. TF12345.TCI	"Topcall/image-blocks"
Unsigned image	TCOSS filename + extension TIF, e.g. TF12345.TIF	"image/tiff;t=4"
Signed message attachment (PDF as an example)	TCOSS file name + ".PDF.PK7"	"application/x-pkcs7-mime"
Signed binary attachment (DOC as an example)	Original file name + ".PK7"	"application/x-pkcs7-mime"
Signed message (PDF with 2D barcode)	TCOSS file name + ".PDF"	"application/pdf"

	TS_FILE_NAME	TS_CONTENT_TYPE
Signed TIF attachment (PDF with 2D barcode)	Original file name, extension "PDF"	"application/pdf"
Original TIF attachment (alternative for PDF with 2D barcode)	Original file name and extension (TIF)	"Topcall/unsigned-TIF"
Signed PDF attachment (ADOBE PDF signature)	Original file name and extension ("PDF")	"application/pdf"
Original PDF attachment (alternative for ADOBE PDF signature)	Original file name and extension ("PDF")	"Topcall/unsigned-pdf"

Error Handling

If adding a digital signature fails, TCDCSIGN terminates the send order negatively and sets error code S1 (document conversion failed). The response field (as visible in TCfW outbox) shows more information.

There are two main reasons why a digital signature cannot be created:

Document too complex for signing (fax signature)

If a document contains too many different fonts, graphics etc, digiSeal server refuses to sign it and produces an error file instead.

TCDCSIGN logs a warning event (ID 6018) with a text similar to the following:

```
Signature server returned error, see file D:\DIGISIGN\Kunde7\OUT
\A_20040921_124331_880.ERR
```

The file mentioned in the event description is the error file created by digiSeal Server.

The response field of the message contains the text "Cannot sign message."

The sender should react by adjusting the document that shall be sent, according to the recommendations in this documentation.

No response from digiSeal Server

This happens if digiSeal Server is not running. Please note that it is necessary to enter the PIN code for the signature card in regular intervals (defined in digiSeal server configuration). This error mostly means that it is time to enter the pin code again.

TCDCSIGN logs a warning event (ID 6019) with a text similar to the following:

```
Timeout waiting for signature server (folder D:\DIGISIGN\KUNDE1\OUT) Please check if
the signature server is running!
```

The response field of the message contains the text "Timeout while signing".

Optional Asynchronous Run Mode

For better performance, TCDCSIGN can run in asynchronous mode. Instead of waiting (for a configurable time) for the signed document, the KCS send order is set to BREAK1 status immediately. A reference to the expected signed file name is stored within the message.

When the send order gets active again (after 1 min, 4 min, 9 min, 16 min, 25 min, 36 min, 49 min, 64 min, 64 min ..) TCDCSIGN checks if the signed output file is ready. This is repeated until all requested signed files for the send order are ready, or until an error occurs.

You enable the asynchronous mode by setting *Sign\WaitTimeSec* to 0.

Note

In asynchronous mode, TCDCSIGN stores several temporary files in the CFG folder (also for email signature).

Fax Signature

This section describes the Fax signature.

Message Format Considerations

The graphical signature created by digiSeal Server is optimized for text data. Signing a message with a high amount of graphics, shaded text or many different fonts may be impossible, as the resulting signature would be too large.

Therefore, it is necessary to

- configure the layout for every invoice type (to sign only important data)
- take into account technical limitations of the solution (concerning overlays and covers)
- change invoice templates where necessary

Configuring the Layout

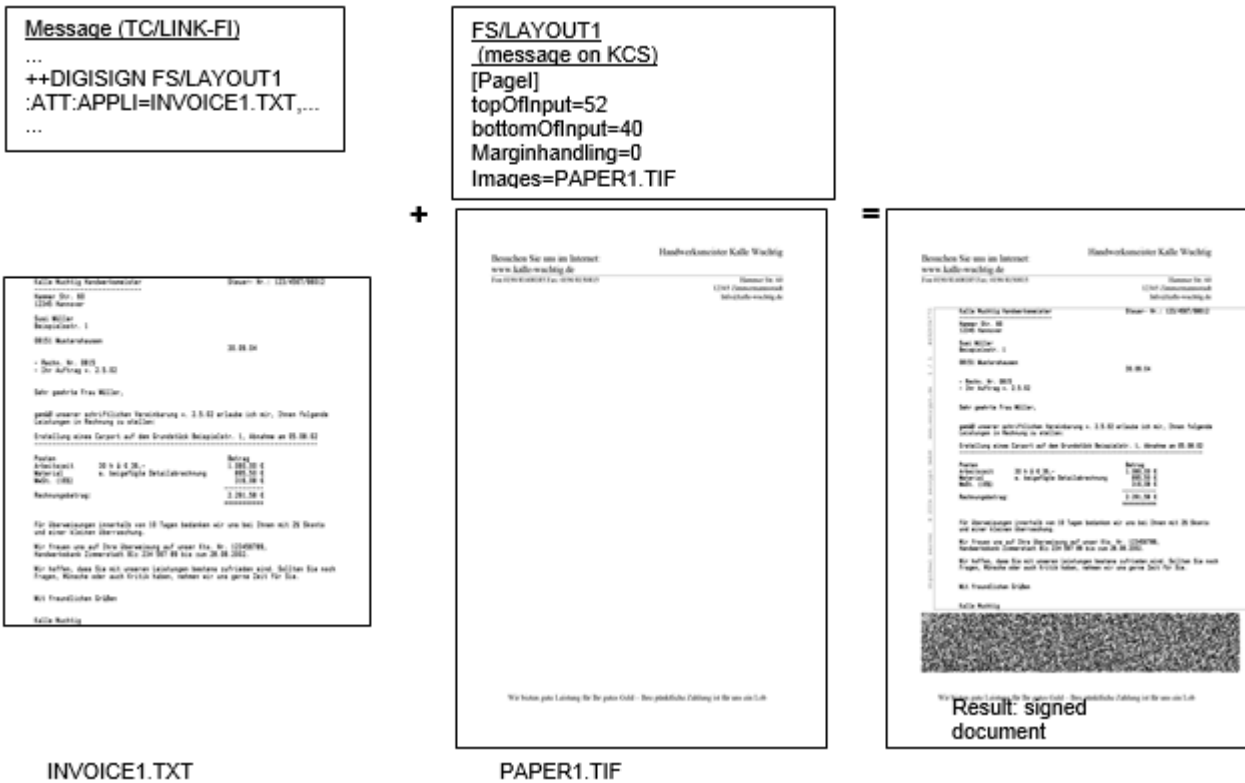
To keep the signature small, only the invoice data shall be signed, but not the header lines, footer lines, company logos etc.

TIF Writing Paper Files Created by the Customer

One way to achieve this is to include only the invoice data in your message, and to put header lines and footer lines into one or more TIF Writing Paper files. DigiSeal server creates a signature for the invoice data, and inserts the header and footer lines afterwards.

Important Writing paper files MUST have resolution 204*196.

Other resolutions are not supported.



In this picture, the original message (posted via TC/LINK-FI) contains a text file without headers and footers.

The ++DIGISIGN directive refers to layout FS/LAYOUT1, which is defined in a document on the TCOSS server. The layout document contains the writing paper that shall be used (PAPER1.TIF).

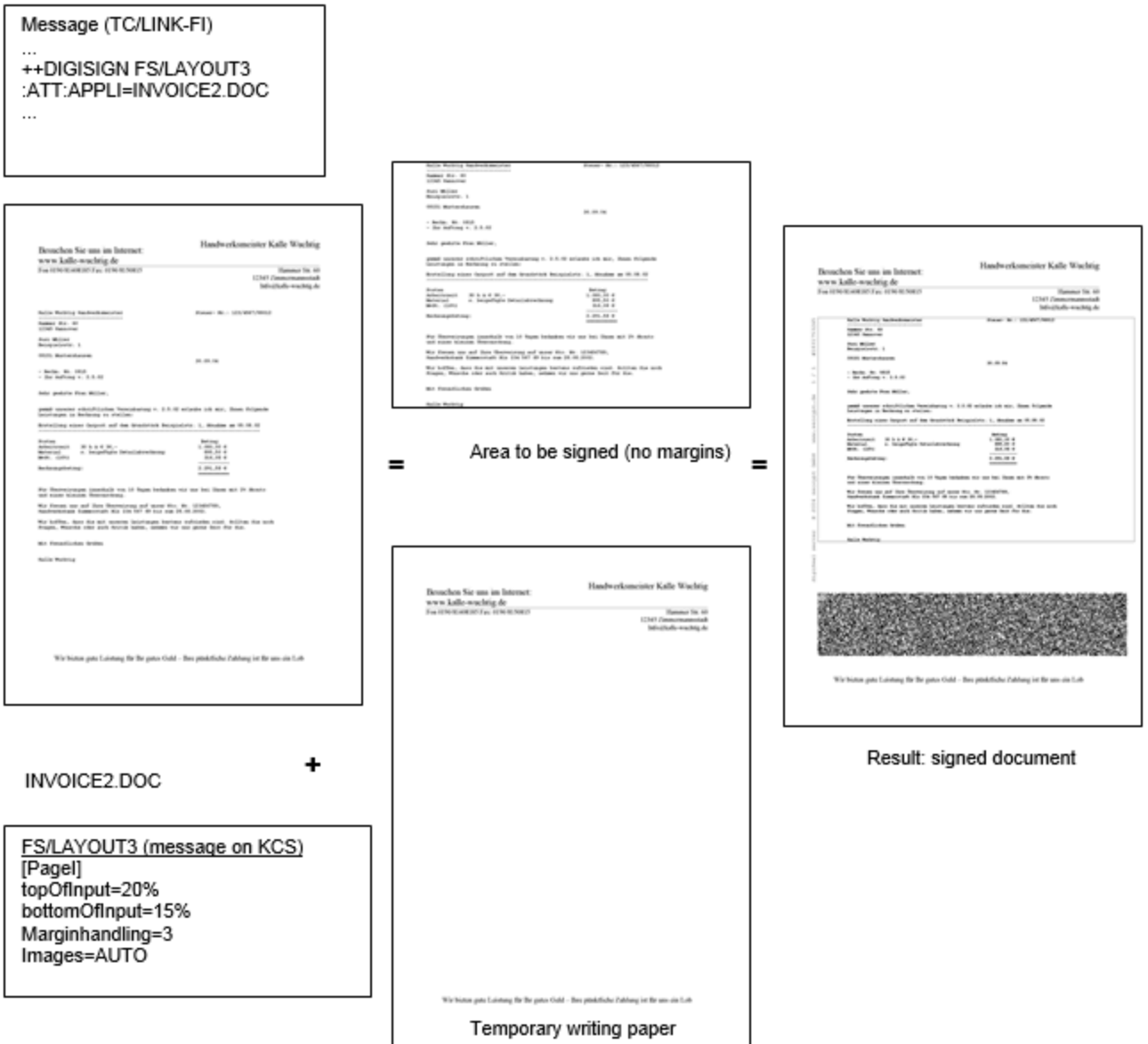
As a result, only the original message content (invoice data) is signed, and the header and footer from the writing paper are added to the message later.

Temporary TIF Writing Paper Files Created Automatically

If it is not possible to remove headers and footers from the original document, TCDCSIGN can separate writing paper information from invoice data automatically, if the length of page headers and footers are configured in the layout document. TCDCSIGN creates a temporary Writing Paper file for the message, and only the area between header and footer is signed.

Automatic Deletion of Empty Space

To avoid scaling of the invoice text, you can tell TDCDSIGN to remove empty space at the top and at the bottom of the invoice data before the signature is created.



In this picture, the original message (posted via TC/LINK-FI) contains a word file **with headers and footers**.

The ++DIGISIGN directive refers to layout FS/LAYOUT3, which is defined in a document on the TCOSS server. The layout document tells TDCDSIGN to **create a temporary writing paper** from the top and bottom margins of the original document, and to **remove empty lines** on top and bottom of the area to be signed.

TCDCSIGN takes the area between header and footer, removes leading and trailing blank pixel lines and then forwards the remaining image to the signature server. The header and footer from the temporary writing paper are added to the message later.

As the leading and trailing blank lines were removed from the input data, less scaling is needed. Removing blank lines has no effect on the size of the barcode.

Layout Definition

In this context, the term layout means instructions given to TCDCSIGN or digiSeal Server that affect the result of the signing process.

Note Old versions of the digiSeal server (from year 2005 or earlier) use a different way of defining layout. You can use the registry value Sign\LayoutSyntax to change the layout syntax. The old syntax is described in older versions of this document or in your digiSeal server documentation.

Instructions for TCDCSIGN:

It is necessary to tell TCDCSIGN which type of preprocessing shall be done before the message is signed (e.g. separating margins from invoice data, deleting empty space) and to specify parameters for these actions (e.g. size of header and footer margin to be cut off). TCDCSIGN must also be told about writing papers to be used.

Instructions for digiSeal Server:

TCDCSIGN forwards some of this information to digiSeal server: which writing paper to use, how much space must be reserved on the resulting page for header and footer.

A customer can define different layouts (if there are different invoice types). The customer's layout information must be stored in the KCS message store. For every layout type there will be one file in the message store.

Which layout to use

Specifying layout type per message:

You can specify the folder/filename of the layout message in the <layout-info> parameter of the +DIGISIGN command.

Using default layout information:

If no layout-info parameter was specified, TCDCSIGN tries to open a default layout message called DIGISIGN in the originator's message folder.

If this message does not exist, TCDCSIGN looks for a DIGISIGN message in the message folder of the TCDCSIGN user.

If this fails also, TCDCSIGN uses a default layout (no margins, empty writing paper), which is defined in file LAYOUT.INI in the customer's interface folder. TCDCSIGN creates this file if needed.

Structure of the layout message

The following parameters must be defined in message text or in an attached text file called LAYOUT.INI. The typical Windows INI-File syntax is used:

```
[PageI]
topOfInput=5%
bottomOfInput=5%
images=LOGO.TIF
MarginHandling=0
```

```
[Page1]
topOfInput=20%
bottomOfInput=5%
Images=LOGOP1.TIF
```

Sections:

Use section [Page1] to specify the default page layout for this document type.

Use other sections, e.g. [Page1],[Page2], [Page1-5] to override the defaults on specific pages.

Mandatory Parameters:

topOfInput: Length (in percent) of the writing paper's header.

bottomOfInput: Length (in percent) of the writing paper's footer.

Images: Name of the writing paper file. This can be the name of a file attachment in the layout message, or the full path name of a file on the TDCDSIGN workstation.

These are TIF 4 files with a resolution of 204*196 DPI (1728*2292 pixel lines).

You can force TDCDSIGN to create a temporary writing paper from the top and bottom data of the first page to be signed. This is achieved by specifying AUTO instead of a writing paper file name. Additionally, the optional parameter *marginhandling* must be 2 or 3 (see below).

Optional Parameters:

MarginHandling: Values 2 and 3 enable preprocessing by TDCDSIGN.

If set to 2, TDCDSIGN removes the top and bottom margins (as specified in parameters *topOfInput* and *bottomOfInput*) from the page before signing.

If set to 3, TDCDSIGN removes empty lines from the top and the bottom of the area to be signed.

LabelAtMargin: Here you can specify the variable part of the margin label for every page section (Page1, Page1 etc.)

Example:

LabelAtMargin=Kofax DigiSign © 2004

Result:

```
Kofax DigiSign (c) 2004      www.secrypt.de      1 / 1      #2697792451
```

Image Resolution

If the faxes to be signed are created via the KCS printer driver (e.g. TDCDSIGN or printing to TCfW), make sure that the printer driver resolution is 204*196.

With other resolutions (e.g. Fine 204 or Fine 200), additional scaling of the image is necessary, - and as a consequence, the matrix code with the signature will be much larger.

Note with TCLINK version 2.11.03 or above, the print resolution for document conversion is automatically set to 204*196 if the message needs a graphical signature.

Technical Limitations Concerning Cover Sheets and Overlays

The signature is created for the graphical image of the invoice data. With standard fax, the fax module is responsible for rendering of text parts of the message, for inserting overlays and evaluating cover sheet variables.

When creating a signature, this rendering is done by TDCSIGN. The image is then signed and the signed image is faxed out as it is.

This has implications on cover sheet usage and TCOSS overlays:

Cover sheet

If the complete message content must be signed, the image created by TDCSIGN contains coversheet and overlays as well.

Most cover sheet variables are evaluated as usual, with the following exceptions:

Variable	Replaced by
\$Docnr\$	Empty string
\$Date1\$, \$Date2\$, \$Time1\$	Date and time when the message was processed by TDCSIGN (local time on TDCSIGN workstation)
Receiver variables	All values taken from first recipient

TDCSIGN evaluates the cover sheet variables on a per-message basis, whereas the fax module would do it separately for every send order.

It is therefore recommended to use only 1 recipient for a fax that must be signed.

Overlays

If the content to be signed is not at the start of the message, any overlays defined in the cover sheet will be inserted later by the fax module. In this case, the cover sheet MUST NOT contain overlays that apply to the area to be signed.

Instead, you must explicitly include the overlay after the ++DIGISIGN line (via ++INC command). Then, it will be part of the signed image.

Adjusting Invoice Content (If Necessary)

Make sure that all necessary information is within the signed area (e.g. tax number, name and address of the company etc.).

If the graphical signature is too big or cannot be created at all, adjust the format of the signed text:

- Do not use many different font types and sizes.
- Use a font with fixed character spacing.
- Do not use graphical elements or shaded areas.
- Bold characters may be better than normal ones.

- Try to keep the amount of lines on a single page small.
- Sans Serif fonts usually make the bar code smaller.

File Interface to digiSeal Server

For fax signatures, the file interface between digiSeal Server consists of 3 directories:

- IN: When requesting a signature, TCDCSIGN places a TIF file (signature relevant data) and an INI file (layout information) to this directory. Both files have the same base name.

digiSeal server removes these files.

- OUT: digiSeal server writes the signed TIF file into this directory. If signature creation fails, an ERR file with error description text is created instead. The signed TIF file and the ERR file have the same base name as the original TIF file.

TCDCSIGN removes the signed TIF file after operation.

Error files are left in the directory (for troubleshooting).

- CFG: holds writing paper files that are referenced in the INI file.

TCDCSIGN removes these files.

A default layout file LAYOUT.INI can be defined per customer (below folders DEFAULT, CUSTOMER1, CUSTOMER2 in this example).

Resulting Message

TCDCSIGN converts the signed TIF file (from folder OUT) to KCS image and stores it in an attachment. The unsigned TIF file (from folder IN) is added as an alternate content.

TCDCSIGN replaces the message part to be signed with this attachment.

The ++DIGISIGN command is removed.

Note The changes are stored in the document on TCOSS. This means that the original content (e.g. the original Word or Excel file) is lost. If you want to keep the original document, you should send the message to a user who has an IN-action to the real recipient. Thus, only the content of the in-action is overwritten, not the original message in the sender's outbox.

The message is then faxed out.

Verifying the Received Fax

If the signed fax is received via a standard fax machine, the recipient needs the following items to verify the graphical signature:

- a TWAIN compatible scanner (minimum resolution 300 dpi)
- the digiSeal reader application (free software available on the internet:
http://www.secrypt.de/downloads/setup_digiSeal_reader.exe)

With the digiSeal reader you can scan in the fax and verify the signature (matrix code integrity, information about the sender's certificate, send time).

The application also shows the decoded invoice information and even allows an optical comparison of the received data and the decoded (original) information. All deviations in the received fax are marked in red color.

If the signed fax is received via a fax server, the recipient needs only the digiSeal reader application. Save the fax image to a TIF4 file. Then you can drag and drop the file onto the digiSeal reader window. digiSeal reader will verify the signature and display the signed content.

Email Signature (PKCS#7)

A signed attachment is encapsulated into a PKCS#7 signed-data file (extension *.pk7). This file contains the original attachment, a digital signature that proves the intactness of the data, and the certificate of the sender.

The recipient can use the digiSeal Reader application to check the signature, view the certificate, and open or extract the original attachment.

You can choose between the following options:

- sign individual file attachments
- create a TIF file containing the complete message (or part of it) and sign this TIF file
- create a PDF file containing the complete message (or part of it) and sign this PDF file

TCDCSIGN knows from the parameter of the ++DIGISIGN command whether an individual message needs a graphical signature (fax) or a digital signature (for the whole image or for each individual attachment).

Note In RTF message text (the default message format for TC/LINK-LN and TC/LINK-MX), the ++DIGISIGN directive is ignored. To create an email signature for a message from Lotus Notes or MS Exchange use service mapping (e.g.: *Sign\Service_NOTES = @MSG_PDF*).

Signature Mode

No layout file is needed. Instead, the parameter of the ++DIGISIGN command is one of the following keywords:

- @ATT: create one digital signature per attachment
- @MSG_TIF: pack all text and image components found after this ++DIGISIGN directive (up to the next ++DIGISIGN directive or to the end of the message) into a TIF file. Then apply a digital signature on this TIF file.
- @MSG_PDF: pack all text and image components found after this ++DIGISIGN directive (up to the next ++DIGISIGN directive or to the end of the message) into a PDF file. Then apply a digital signature on this PDF file.

Implications:

- When using the keywords @MSG_TIF or @MSG_PDF, all components of the area to be signed should have an image or text representation, otherwise they will not be part of the signed TIF or PDF file.

When creating the message via TC/LINK, you must make sure that document conversion occurs (e.g. by sending to a service that does not support binary attachments, or via registry values *General\AltForceBinToTci* and *General\AltForceImgToTci*).

- If @MSG_TIF or @MSG_PDF is encountered at the start of the message, the signed TIF or PDF file will also include the cover page.

Examples:

The following transaction files are posted via TC/LINK-FI.

Example 1:

The message contains text and attachments. The result shall be a signed PDF file.

```
FROM: SN=someone,SE=TCFI,N=someone
TO: SN=mailrecipient,SE=SMTP,N=annemarie.fischer@kofax.com
++DIGISIGN @MSG_PDF
Please find attached an important documentation.
:ATT:NAME=IMPORTANT.DO,APPLI=IMPORTANT.DOC
```

The recipient gets an attachment with extension “.PDF.PK7”. He can use digiSeal Reader to open this file, verify the signature and extract a PDF file with the complete message content.

Example 2:

The message contains text and 2 attachments. The result shall be a signed PDF file for the first part of the message (text and DOC file) and a signed Excel document.

```
FROM: SN=someone,SE=TCFI,N=someone
TO: SN=mailrecipient,SE=SMTP,N=annemarie.fischer@kofax.com
++DIGISIGN @MSG_PDF
Please find attached an important documentation.
:ATT:NAME=IMPORTANT.DO,APPLI=IMPORTANT.DOC
++DIGISIGN @ATT
:ATT:NAME=SHEET1.XL,APPLI=SHEET1.XLS
```

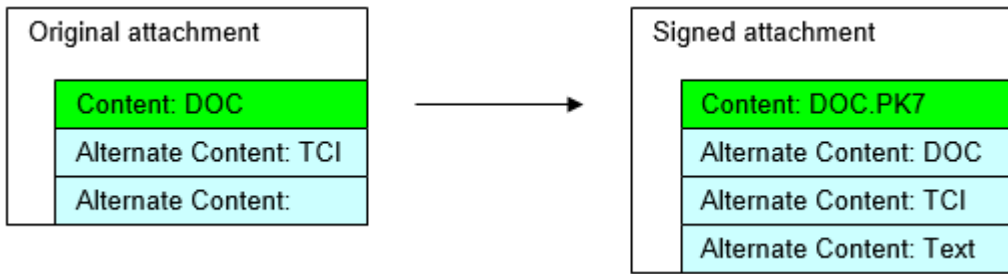
The recipient gets two attachments: one with extension “.PDF.PK7”, and one with extension “.XLS.PK7”. He can use digiSeal Reader to open the files, verify the signature and extract the file content.

File Interface to digiSeal Server

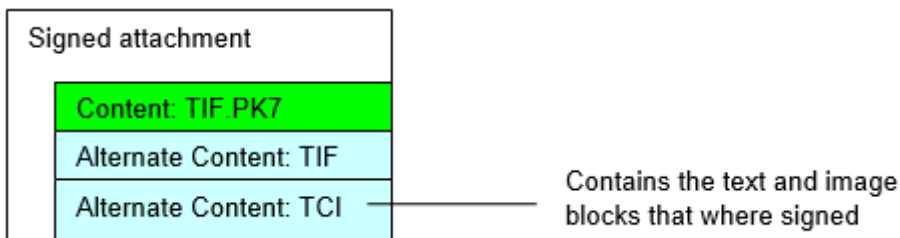
Folder names used for PKCS#7 email signature are BIN_IN and BIN_OUT. Like with fax signature, it is possible to have several customers (with separate folders) per TCOSS instance.

Resulting Message

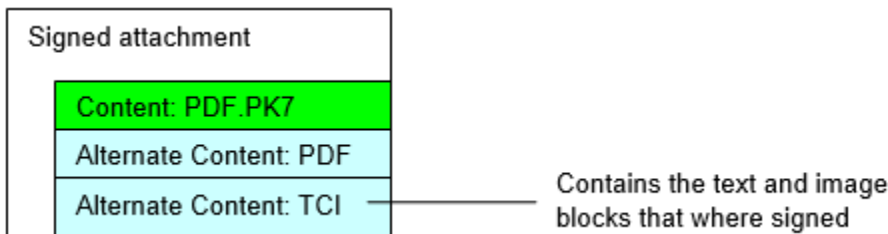
Layout of an attachment signed with ++DIGISIGN @ATT directive:



Layout of a signed attachment created with the ++DIGISIGN @MSG_TIF directive:



Layout of a signed attachment created with the ++DIGISIGN @MSG_PDF directive:



Email Signature (PDF with 2D Barcode)

The result of this signature is a PDF file with a 2D barcode, for printout on a laser printer or an inkjet printer. The barcode is similar to the fax signature barcode, but the resolution is higher. The barcode holds the document data, the certificate of the sender and a signature.

The recipient can use the digiSeal Reader application to check the signature, view the certificate, and open or extract the original attachment.

You can choose between the following options:

- sign individual TIF file attachments
- create a TIF file containing the complete message (or part of it) and sign this TIF file

TCDCSIGN knows from the parameter of the ++DIGISIGN command whether an individual message needs a graphical signature (fax) or a digital signature (for the whole image or for each individual attachment).

Note In RTF message text (the default message format for TC/LINK-LN and TC/LINK-MX), the ++DIGISIGN directive is ignored. To create an email signature for a message from Lotus Notes or MS Exchange use service mapping (e.g.: *Sign\Service_NOTES = @MSG_2D*).

Signature Mode

No layout file is needed. Instead, the parameter of the ++DIGISIGN command is one of the following keywords:

- @ATT_2D: create one digital signature per TIF attachment (black and white TIF files only).
- @MSG_2D: pack all text and image components found after this ++DIGISIGN directive (up to the next ++DIGISIGN directive or to the end of the message) into a TIF file. Then create a signature for this TIF file.

Implications:

- When using the keyword @MSG_2D, all components of the area to be signed should have an image or text representation, otherwise they will not be part of the signed result file.

When creating the message via TC/LINK, you must make sure that document conversion occurs (e.g. by sending to a service that does not support binary attachments or via registry values *General\AltForceBinToTci* and *General\AltForceImgToTci*).

- If @MSG_2D is encountered at the start of the message, the signed result file will also include the cover page.
- Result files created with @MSG_2D have no better resolution than Fax fine mode, as the input file is created from the KCS fax image format.
- Use the keyword @ATT_2D to sign input files with a resolution higher than 204*196 dpi.

The input TIF file is given directly to digiSeal Server, so there is no loss by converting to KCS fax image.

Examples:

The following transaction files are posted via TC/LINK-FI.

Example 1:

The message contains text and an attachment. The result shall be a signed PDF file with 2D barcode.

```
FROM: SN=someone,SE=TCFI,N=someone
TO: SN=mailrecipient,SE=SMTP,N=annemarie.fischer@kofax.com
++DIGISIGN @MSG_2D
Please find attached an important documentation.
:ATT:NAME=IMPORTANT.DO,APPLI=IMPORTANT.DOC
```

The recipient gets an attachment with extension “.PDF”.

Example 2:

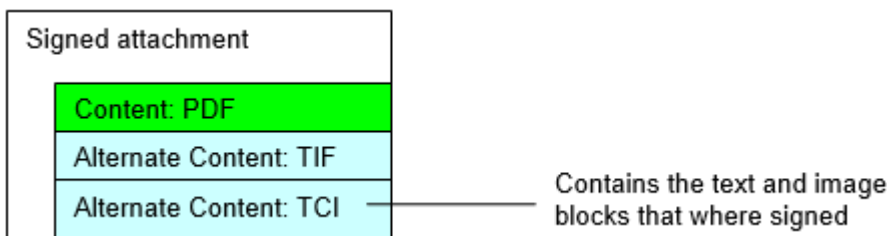
The message contains text and a TIF attachment. The result shall be a signed PDF file for the attachment.

```
FROM: SN=someone,SE=TCFI,N=someone
TO: SN=mailrecipient,SE=SMTP,N=annemarie.fischer@kofax.com
Please find attached an important documentation.
++DIGISIGN @ATT_2D
:ATT:NAME=IMPORTANT.TI,APPLI=IMPORTANT.TIF
```

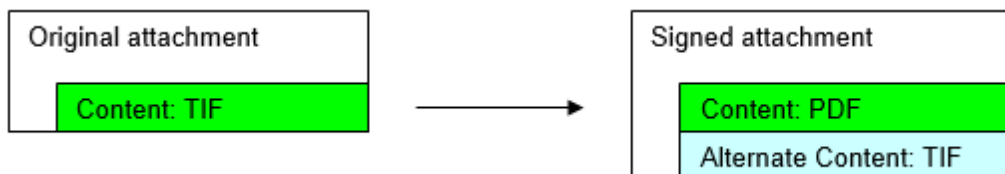
The recipient gets an attachment with extension “.PDF”.

Resulting Message

Layout of an attachment signed with ++DIGISIGN @MSG_2D directive:



Layout of an attachment signed with ++DIGISIGN @ATT_2D directive:



Note If an additional TCI alternative content exists in the original attachment, it is also part of the signed attachment.

Email Signature (Adobe PDF)

The signature corresponding to the ADOBE PDF standard version 1.6 is embedded into the PDF file.

Only PDF attachments are converted into signed PDF attachments. Input files with a different format must be converted to PDF via third party applications (e.g. a PDF printer).

The input PDF file is given directly to digiSeal Server, no resolution loss by converting to KCS fax image.

Acrobat Reader (version 6.0 or above) evaluates and displays signature details on its Signature tab. No additional client software is needed for verification.

Signature Mode

This signature mode is requested via the text directive `++DIGISIGN @ATT_PDF`.

All PDF files attachments after the text directive will be signed.

Note In RTF message text (the default message format for TC/LINK-LN and TC/LINK-MX), the `++DIGISIGN` directive is ignored. To create an email signature for a message from Lotus Notes or MS Exchange use service mapping (e.g.: `Sign\Service_NOTES = @ATT_PDF`).

Example:

The following transaction file is posted via TC/LINK-FI.

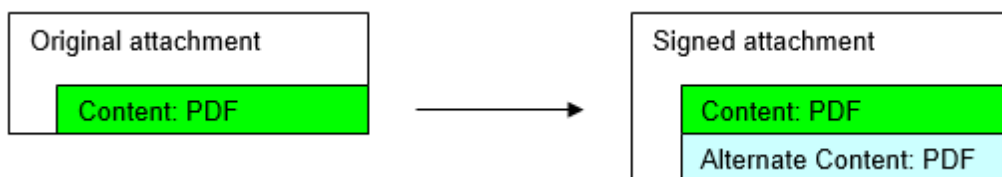
The message contains text and attachments. Every PDF attachment shall get an ADOBE PDF signature.

```
FROM: SN=someone,SE=TCFI,N=someone
TO: SN=mailrecipient,SE=SMTP,N=annemarie.fischer@kofax.com
++DIGISIGN @ATT_PDF
Please find attached important documentation files.
:ATT:NAME=IMPORTANT1.PD,APPLI=IMPORTANT1.PDF
:ATT:NAME=IMPORTANT2.PD,APPLI=IMPORTANT2.PDF
```

The recipient gets two attachments with extension “.PDF” and can verify their signature with Acrobat Reader.

Resulting Message

Conversion of an attachment via `++DIGISIGN @ATT_PDF` directive:



Notes:

The alternative content holds the unsigned (original) PDF file. This alternative can be suppressed via TDCSIGN configuration (registry value `Sign\PdfKeepUnsigned`).

If an additional TCI alternative content exists in the original attachment, it is also part of the signed attachment.

Chapter 4

XML Attachment Parsing

This feature has been developed to import XML data to Kofax Capture via TWS and KCS Capture Connector. It is currently only useful in this context.

TWS expects that the message header holds information about the first XML attachment found in a message. All messages posted by TCLINK already have this information.

But for messages created via other applications (TWS, TCfW, standalone PS applications), XML attachment parsing can only be done via the TCOSS converter.

See also *TC/LINK Technical Manual* for additional information.

Overview of Operation

Messages submitted to TCOSS can be marked as needing XML attachment parsing.

TCOSS forwards such messages to an internal queue that is polled by the TCOSS Message Preprocessor .

TCDCSIGN uses a configurable DLL (XML Exit) for this feature. Thus, the default implementation provided by the product can easily be replaced with a module written by PS or by a customer.

XML Exit Configuration

Two registry values below the TCDCSIGN subkey define the location of the XML exit and its configuration file.

Name	Type	Initialized by Setup to	Description
XML\ExitDLL	SZ	C:\TCOSS\SYSTEM\XmlExit.dll	Path name of Exit DLL
XML\ExitConfig	SZ	C:\TCOSS\SYSTEM\ XmlExitConfiguration.xml	Path name of configuration file

If *XML\ExitDLL* is empty or does not exist, the XML parsing feature is disabled. In this case, messages can still be routed to the TCOSS Message Preprocessor queue, - but the XML attachments will not be parsed and no additional information will be written into the message header.

The default configuration provided in *XmlExitConfiguration.xml* is:


```

<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2013 (x64) (http://www.altova.com)-->
<XMLExit xsi:noNamespaceSchemaLocation="XMLExit.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ExtensionTypes>text/xml, .xml, text/oxi, .oxi</ExtensionTypes>
  <ConfiguredServices>
    <SenderService></SenderService>
    <RecipientService></RecipientService>
  </ConfiguredServices>
  <SchemaMustExist>DISABLED</SchemaMustExist>
  <SchemaPath></SchemaPath>
  <RejectMessage>>false</RejectMessage>
  <EmbeddedMessage>>false</EmbeddedMessage>
</XMLExit>

```

You can change the configuration settings via a text editor or XML editor.

XML node name	Description	Additional remarks
ExtensionTypes	This comma separated list defines which attachments are regarded as XML. The list can hold MIME content types and file extensions (with leading dot).	Content type information is only available in messages received via SMTP based links.
ConfiguredServices	Can be used to restrict XML parsing based on sender or recipient service. Holds two comma separated lists of KCS service names. The XML Exit compares the first active address of sender and active recipients against these lists. XML parsing is only done if the sender address and at least one recipient address match the configured services. If a list is empty, any service matches. If a list contains one or more values, only the service names from the list match.	In the default configuration, both lists are empty and no service based restrictions are done.
SenderService	List of sender services	
RecipientService	List of recipient services	
SchemaMustExist	Controls the optional validation against a schema. The following values are possible: DISABLED - validation is disabled OPTIONAL – validation is done if schema exists REQUIRED – validation is necessary	The location of schema files is defined in SchemaPath.
SchemaPath	Full path name of a directory holding schema files and sample XML files.	
RejectMessage	Defines how to handle XML processing errors. Possible values: true – send order is terminated negatively on error false – send order is sent, but contains no header information about XML attachment	

XML node name	Description	Additional remarks
EmbeddedMessage	Defines how to handle XML attachments in embedded messages. Possible values: true – XML processing is done recursively (embedded message also holds information about its first XML attachment) false – embedded messages are not parsed for XML information	

Schema Validation

The optional schema validation feature relies on XML schema files (.xsd) and XML sample files (.xml).

The directory configured as SchemaPath must have a subfolder for each type of XML content that shall be validated. The name of the subfolders is not relevant, but each subfolder must contain two files with fixed names: sample.xml (a sample XML file) and schema.xsd (the schema).

Chapter 5

Maintenance

This section describes the maintenance information.

Registry Keys

The following registry values are used by TCDCSIGN. They are below *HKLM\Software\TOPCALL\TCDCSIGN*.

Subkey TOPCALL:

Name	Type	Default	Description
PollFlags	DWORD	0x1400	Bit field. hex 400 – signature mode only hex 1000 – XML mode only hex 1400 – signature and XML parsing
MaxRetries	DWORD	10	Maximum number of logon retries after loss of connection to KCS
PassWord	SZ		Encrypted password of user
Path	SZ		TC server path
PollCycle	DWORD	30	Poll cycle in seconds
RetryTime	DWORD	30	Time (in seconds) till first logon retry after loss of connection to KCS
Server	SZ		TC server name
TCSIDebug	DWORD	0	If 1, messages are traced before conversion
User	SZ		TC user used by TCDCSIGN

Subkey GENERAL:

Name	Type	Default	Description
MailDebug	DWORD	0	If 1, messages are traced after conversion
TraceLevel	DWORD	1 (trace errors only)	Trace level

Subkey DEBUG:

A subkey Debug is used by the developer (for test and troubleshooting):

Name	Type	Default	Description
Debug\ SignSimulator	SZ		Full path name of simulator program, e.g. C:\TOPCALL\SHARED\SFS.EXE
Debug\ LeaveSignatureFiles	DWORD	0	Only for testing with the simulator program. If set to 1, the files in the interface directories are not deleted.

Subkey SIGN:

A subkey Sign holds configuration parameters for signature creation.

Name	Type	Default	Description
Sign\ DefaultLayoutMsg	SZ	DIGISIGN	Name of default layout message
Sign\ DefaultCustomer	SZ	DEFAULT	Default customer name (for directory mapping)
Sign\ RootDir	SZ	C:\DIGISIGN	Common root directory
Sign\ DirFaxIn	SZ	IN	Input folder for fax signature
Sign\ DirFaxOut	SZ	OUT	Output folder for fax signature.
Sign\ DirCfg	SZ	CFG	Writing paper folder for fax signature.
Sign\ DirBinIn	SZ	BIN_IN	Input folder for binary (PK7) signature.
Sign\ DirBinOut	SZ	BIN_OUT	Output folder for binary (PK7) signature.
Sign\ Dir2Din	SZ	2D_IN	Input folder for 2D signature (printer)
Sign\ Dir2Dout	SZ	2D_OUT	Output folder for 2D signature (printer)
Sign\ DirPdfIn	SZ	PDF_IN	Input folder for ADOBE PDF signature
Sign\ DirPdfOut	SZ	PDF_OUT	Output folder for ADOBE PDF signature
Sign\Dir\ <CustomerName>	SZ		Interface directory for customer <CustomerName>

Name	Type	Default	Description
Sign\ CustomerField	SZ		If several customers share the same TCOSS instance, this configuration value tells TDCDSIGN which field in the user profile holds the customer name. Possible values: Company Department Freetext Salutation Fullname ID Group CostCenter Location
Sign\ WaitTimeSec	DWORD	10	Maximum wait time for signed fax image. If 0: asynchronous mode
Sign\ ErrExtension	SZ	ERR	Extension of error file produced by signature server.
Sign\ InstanceNum	DWORD	1	Instance number (1 to 26). Determines first character of TIF file names (1 = A, 2 = B etc.)
Sign\ LayoutSyntax	DWORD	1	0 – old syntax for layout definition (2005 or earlier) 1 – new syntax
Sign\ PdfKeepUnsigned	DWORD	1	Only for ADOBE PDF signature. Possible values: 0: original PDF attachment is not included in result message 1: original PDF attachment is stored as alternative content.
Sign\ Service_<name>	SZ		Must be created manually. <name> is the service name. The value of this key is the signature parameter to be used, e.g.: @ATT, @ATT_2D etc.

Subkey XML:

Subkey XML holds configuration parameters for XML parsing.

Name	Type	Initialized by Setup to	Description
XML\ ExitDLL	SZ	C:\TCOSS\SYSTEM\XmlExit.dll	Path name of Exit DLL
XML\ ExitConfig	SZ	C:\TCOSS\SYSTEM\ XmlExitConfiguration.xml	Path name of configuration file

Registry values below *Sign\Dir* and registry values *Sign\Service_<name>* must be created manually. All other registry values are created automatically when TDCDSIGN starts or during Setup.

Event Log Entries

Code	Severity	Description	Corrective Action	Parameter Description
6000	W	Error reading configuration from registry	Check if registry subkey TCDC exists, check trace file for names of missing values	
6002	E	cannot create stop event	Check available NT resources.	
6003	E	cannot start RPC communication	Check version of TCRPC32.DLL	
6004	E	stopped by TCSRVR (no real error)		
6005	E	no conversion needed, message is already converted		
6006	E	Error opening temp. file	Check trace file for path of temp. file, check if TCDCSIGN has read and write permissions in this directory	
6E07	E	Error reading data from temp. file	Trace file contains temp. file path. Check access rights to this path.	
6009	E	Error finding path for temp. files (environment var "TMP" or "TEMP")	Trace file should contain more information.	
6010	E	error creating temp. file name		
6012	E	unknown TCSI object encountered	Send TCSI-level trace to developer.	
6013	W	No valid license for %1. Messages are not converted.	Obtain a license from Kofax staff.	%1: requested license type
6016	I	TCDC version %1 started		&1 Version number
6018	W	Signature server returned error, see file %1		%1: error file in interface directory
6019	W	Timeout waiting for signature server (folder %1)	Check if signature server is running.	%1: output directory (where signed file is expected)

Trace File

TCDC.EXE writes trace output to C:\TCOSS\TRACE\TCDCxx.TRC.

The following TCDCSIGN trace levels are used:

Range	Meaning
0	No trace output
0x01 – 0x07	Errors only
0x08 – 0x0F	+ warnings
0x10 – 0x1F	+ major function calls + returns
0x20 – 0x3F	+ major subroutine calls + returns
0x40 – 0x7F	+ all minor routine calls + returns
0x80 – 0xFF	+ debug and test outputs
0x100 and above	reserved