

Kofax Communication Server

TOM Technical Manual

Version: 10.2.0



© 2018 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.

CONTENTS

1. Overview	4
1.1 About This Manual	4
1.2 What Is the Kofax Communication Server?	4
1.3 Why Should I Connect My Application to KCS?	4
1.4 What Do I Need for Connecting to KCS?	4
2. Getting Started.....	5
2.1 Related Documentation	5
2.2 Prerequisites	5
2.2.1 KCS Environment	5
2.2.2 Interface Requirements	5
2.3 Sending Simple Messages to KCS.....	6
2.4 Defining the Originator	7
2.5 Enhanced Message Parameters	8
2.6 Attaching Documents, Binaries and Similar	9
2.7 Getting Notifications from KCS	10
2.8 Matching Notifications to Original Messages.....	11
2.9 Receiving Messages from KCS	12
2.10 Optional: Sending Notifications to KCS	13
3. Advanced Hints	14
3.1 Supported Code Pages	14
3.2 Addressing To X.400 Address.....	14
3.3 Addressing To KCS Address Book Entries	14
3.4 Using CR/LF as Parameter Delimiter	15
3.5 Directory Synchronization	15
3.5.1 Dirsync within the Transaction File	15
3.5.2 Dirsync with an Attached CSV File	16
3.6 All-for-one Key System Support.....	19
3.7 Syntax Errors in Messages to KCS.....	23
4. Syntax Reference.....	25
4.1 TC Open Message Format Structure Overview.....	25
4.2 General Message Parameters.....	26
4.2.1 Default Handling of Sending Parameters.....	27
4.3 The Sender Object (FROM:)	27
4.4 The Recipient Objects (TO, CC, BCC, AUTH).....	28
4.5 Sender/Recipient Addressing	28
4.5.1 Standard Format for Sender/Recipient Address.....	28
4.5.2 Shortcut Format for Recipient Address.....	29
4.6 Banking Key Information (KEYINFO:)	29
4.7 The Correlation Object (NFINFO:)	30
4.8 CSV File Information for Dirsync (CSVINFO:)	30
4.9 The Start of Message Content (TXT:).....	30
4.10 Attached Files (:ATT:).....	31
4.11 Attached KCS Objects (:INC:)	31
4.12 Authorizing Notifications (from All-for-One server).....	31
4.13 Important Notes on TC Open Message Format Syntax	32
5. Appendix.....	33
5.1 Registry Values Used By TC Open Message Format	33

1. OVERVIEW

1.1 About This Manual

This document is intended for developers of messaging applications that want to connect to the Kofax Communication Server.

It consists of three major parts:

- The “Getting Started” section helps for a short investigation of the Kofax Communication Server File API capabilities.
- The “Advanced Hints” gives possibilities for fine-tuning your application.
- The “Syntax Reference” gives a detailed summary of all available parameters.

1.2 What Is the Kofax Communication Server?

The powerful Kofax Communication Server ties together almost any kind of email, plus the most important public services like Fax and Telex.

For example, a user on Microsoft Exchange can send a message to a user on Lotus Notes, and vice versa. The same applies for fax, telex, Internet mail, X.400, SAP, cc:Mail, HP OpenMail, and some more.

But, even better, you can enable support of all these services for your application by using a simple, open interface – the Open Message Format (TOM).

All of these connections are available and addressed as so-called “**services**”.

1.3 Why Should I Connect My Application to KCS?

Of course, you can write your gateway to the Internet, or MS Exchange and Lotus Notes yourself. But, maybe you would prefer a solution that is ready to serve almost any email product, plus public services.

Kofax develops integrations to new mail systems, as well as keeps track of existing ones. So, using the File API will automatically keep your application “up-to-date”; you don’t have to worry about new releases.

1.4 What Do I Need for Connecting to KCS?

With Kofax Communication Server, we deliver turn-key solutions.

Basically, all you need is to select the services you want to use; Kofax will install the whole system, and configure it according to your (and your customers) needs.

To connect you application to KCS, you need to choose one of the transports where the Open Message Format is available. At the moment, there are two of them:

- The TC/LINK-FI which supports a file-based interface
- The TC/LINK-MQ which connects to IBM MQSeries.

2. GETTING STARTED

This section will help you to get a quick overview of the Open Message Format.

2.1 Related Documentation

This document is not intended to be a reference of all features available from the Kofax Communication Server. If you need a full reference of all features, refer to the following documents:

- TC/LINK-FI Manual (the product that brings the open message format to a file interface)
- TC/LINK-MQ Manual (the product that brings the open message format to IBM MQSeries environment)
- TCLINK Manual (the main component for all KCS email services)
- TCOSS System Manual (for configuration of the TCOSS server, and fax/telex interfaces)

2.2 Prerequisites

2.2.1 KCS Environment

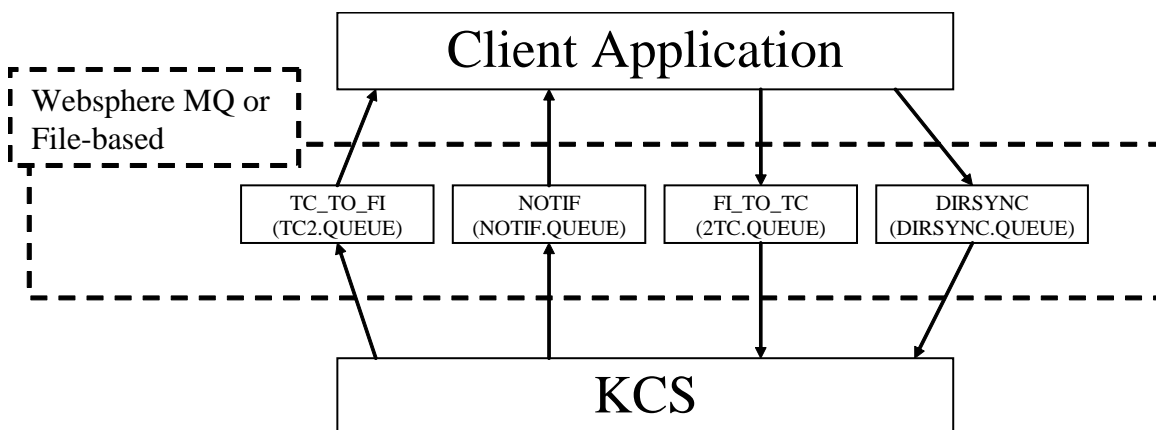
This document makes some assumptions about the installed environment.

- KCS server
- Fax channel
- TC/LINK-SM
- TC/LINK-FI or TC/LINK-MQ

2.2.2 Interface Requirements

If the KCS environment is installed properly, there are four directories (TC/LINK-FI) or queues (TC/LINK-MQ) that handle all the functionality of the TC Open Message Format:

- In the **FI_TO_TC** directory (the **2TC.QUEUE**), you have to put all message or notification files you want to be sent via KCS to any of its services.
- The **NOTIF** directory (the **NOTIF.QUEUE**) gives the notification files for the messages sent via FI_TO_TC.
- **TC_TO_FI** (the **TC2.QUEUE**) holds all incoming messages (e.g. from FAX).
- The **DIRSYNC** directory (the **DIRSYNC.QUEUE**) can be used to maintain the user store on KCS.



2.3 Sending Simple Messages to KCS

All things discussed so far were rather theoretic. To make life easier, let's start with a simple example: If you generate the following text file ("test.txt"):

```
TO: SERVICE=FAX, NUMBER=66133831
TXT:
Hello there!

I am currently investigating the TC Open Message Format for use in my application.
```

, and put it to the FI_TO_TC directory (...the 2TC.QUEUE), you will get a fax sent to the number "66133831", containing of the text given from the "TXT:" keyword onwards:

```
Hello there!

I am currently investigating the TC File API for use in my application.
```

In this example, you can see how the basic addressing works:

Every address consists of a

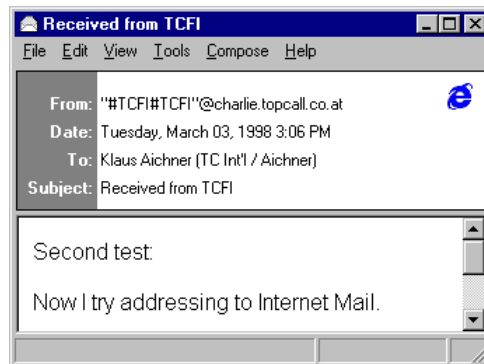
- SERVICE ... indicating the destination mail system, and a
- NUMBER ... indicating the recipient on the destination mail system.

These SERVICE / NUMBER pairs can be used for almost every destination. Just one more example:

```
TO: SERVICE=SMTP, NUMBER=AI@kofax.com
TXT:
Second test:

Now I try addressing to Internet Mail.
```

This message will go to SMTP (Internet mail), and may look as follows when received in a client (Outlook Express in the example):



Well ... the text looks as expected; but where does the message originator (the "From:") come from? And – even more – you may want to set the originator address via the TC Open Message Format, so that replies from the Internet mail client will work!

Come and see how easy this can be done ... in the next section.

2.4 Defining the Originator

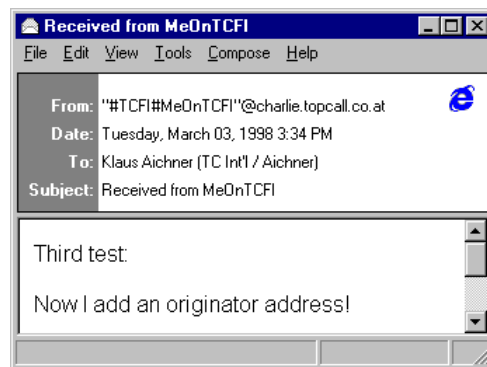
In the previous examples, we did not specify an originator. That is why a configured default originator was inserted to the message.

Defining the originator address in the message is similar to giving the recipient – you have to insert a “FROM:” line with a SERVICE/NUMBER address pair, additionally you have to set the SNAME parameter to the user-id.

```
FROM: SERVICE=TCFI, NUMBER=MeOnTCFI, SNAME=MyShortName
TO: SERVICE=SMTP, NUMBER=AI@kofax.com
TXT:
Third test:

Now I add an originator address!
```

Note: The service “TCFI” identifies the TC File Interface service! Therefore, you can address to the TC File API from any connected system by using the service “TCFI”. If you are using the MQSeries interface, this would have been “SERVICE=MQ”.



If the Internet mail client user presses “Reply”, then the reply message will go to the TC_TO_FI directory (the TC2.QUEUE) – and will look like

```
SUBJECT="Re: Received from MeOnTCFI"
FROM: SERVICE=SMTP, NUMBER=AI@kofax.com, ACTIVE=YES
TO: SERVICE=TCFI, NUMBER=MeOnTCFI, ACTIVE=YES
TXT:
Reply!

-----Original Message-----
From: "#TCFI#MeOnTCFI"@charlie.kofax.com <"#TCFI#MeOnTCFI"@charlie.kofax.com>
To: Klaus Aichner (TC Int'l / Aichner) <AI@kofax.com>
Date: Tuesday, March 03, 1998 3:31 PM
Subject: Received from MeOnTCFI

Third test:

Now I add an originator address!
```

You already discovered the “SUBJECT” field. Now it’s time to add some more message properties (all of them are optional; but they may be useful!)

2.5 Enhanced Message Parameters

Strictly spoken, there are two types of enhanced parameters:

- Per-message parameters (must be given before the Originator information)
- Per-recipient parameters (must be given for every TO: section).

A few self-explaining examples for per-message parameters:

```
SUBJECT="This Is My First Message Subject In TC Open Message Format"
FROM: SERVICE=TCFI, NUMBER=MeOnTCFI
TO: SERVICE=SMTP, NUMBER=AI@kofax.com
TXT:
Looks good, doesn't it?
```

```
DATE=20000101, TIME=010000
FROM: SERVICE=MQ, NUMBER=MeOnMQSeries
TO: SERVICE=SMTP, NUMBER=AI@kofax.com
TXT:
This message will be sent on Jan. 1st, 01:00 AM.
No need to say that Kofax is fully Year-2000 compliant!
```

```
P=LOW
FROM: SERVICE=TCFI, NUMBER=MeOnTCFI
TO: SERVICE=FAX, NUMBER=6613321
TXT:
This message has low priority. Therefore, the fax will be sent at low-tariff time (after
midnight)
```

One more example for per-recipient options (note that you can specify multiple recipients in a single message!)

```
FROM: SERVICE=TCFI, NUMBER=MeOnTCFI
TO: P=HIGH, SERVICE=SMTP, NUMBER=Hurry@Urgent.co.at
TO: P=LOW, SERVICE=SMTP, NUMBER=LazyBoy@sleep.com
TXT:
Sending to multiple recipients with different priority setting ...
```

You may say: "That's very nice, but it is quite annoying that I can only send plain-text messages! I thought it's the time of electronic documents and multimedia ...".

You are perfectly right. And the TC Open Message Format offers you some possibilities to include binaries of any kind, images, audio or even video files into your messages.

2.6 Attaching Documents, Binaries and Similar

Again, this is best shown with an example:

```
SUBJECT="TC Open Message Format description"
FROM: SERVICE=MQ, NUMBER=MeOnMQSeries
TO: SERVICE=FAX, NUMBER=66133831
TXT:
Thanks for your interest in the Kofax solution.
Enjoy one of the last adventures ... connect to Kofax!

Bye,
Klaus
:ATT: NAME=1, APPLICATION=TOM_21902.doc
```

You see the :ATT: section ... it indicates that there is a document attached. This example is for attaching from MQSeries; below you find one for the File interface.

The two parameters following the :ATT: describe the attached document:

- The "NAME" gives the physical attachment name (for TC/LINK-FI, this must be including the full path; for TC/LINK-MQ, it gives the relevant part of the CorrelId)
- The "APPLICATION" indicates the file type (.DOC -> this is a Word Document)!

Note:

- On TC/LINK-FI, do NOT put the attachment files in the FI_TO_TC directory if they have a three-character extension! The three-character extension is reserved for the text files (also called "transaction files")!
- After successful sending, the attachment file is removed from the directory/queue (as also the transaction file is).
- To make sure that all attachments are available when the transaction file is opened, you must put ALL attachments before putting the transaction file!

Now, what happens to that word attachment? It isn't good to send binaries via a Fax service ...

At this point, the TC/LINK Document Converter comes into the game. It will convert the Word attachment into a Fax document – just like if you print the document, and put the paper into a physical fax machine.

But you don't need to care about that all ... it runs fully automatic. Simply attach the file and KCS will do the rest.

2.7 Getting Notifications from KCS

In the last chapter, we sent the TOM Technical Manual to a friend's fax machine. But how can you be sure that it has been delivered correctly?

Well – the default is: If something went wrong, you get a non-delivery notification into your NOTIF directory (or, on MQSeries, your NOTIF.QUEUE). But you can also request successful delivery notifications by inserting a single parameter into your transaction file:

```
SUBJECT="TC Open Message Format Description", NF=ALL
FROM: SERVICE=TCFI, NUMBER=MeOnTCFI
TO: SERVICE=FAX, NUMBER=66133831
TXT:
Thanks for your interest in the Kofax solution.
Enjoy one of the last adventures ... connect to Kofax!

Bye,
Klaus
:ATT: NAME=C:\TEMP\TOM_21902.doc, APPLICATION=TOM_21902.doc
```

This "NF=ALL" will tell KCS to send a notification back as soon as delivery was done. In the NOTIF directory, you will find a file telling something like

```
SUBJECT="Del: TC Open Message Format Description ", TYPE=NOTIF, MCORR=00001639982
FROM: ACTIVE=YES, SERVICE=FAX, NUMBER=66133831
TO: ACTIVE=YES, SERVICE=TCFI, NUMBER=MeOnTCFI
NFINFO: STATUS=DEL, TIME=980303:163100, MCORR=00001639982
TXT:
TOPCALL Delivery Notification
-----

Message 00001639982 successfully sent to FAX,66133831
Time sent: 98-03-03 16:31:00
Subject: TC Open Message Format Description
Cost: 5 for GUEST
```

I told "something like" because these notifications can be configured for different languages and different information level. The most important things are

- "TYPE=NOTIF" tells you that this is a notification
- The "NFINFO:" section tells you if delivery was successful ("STATUS=DEL"), and when the document was delivered (TIME=YYMMDD:hhmmss).

Usually, you will get some more parameters (like KCS "last action" codes, Correlation Information); the general rule is: only take what you need, ignore the rest!

2.8 Matching Notifications to Original Messages

As long as you send messages one by one, there is no problem in finding out which notification is correlating to which message. But, as soon as all runs automatically (with a throughput of several thousand messages per hour), there is the need for providing some correlation information!

This can be done in the TC Open Message Format – see the following example:

```
SUBJECT="TC Open Message Format Description", NF=ALL
FROM: SERVICE=MQ, NUMBER=MeOnMQSeries
TO: C1=4711, C2=MyFirstMessageWithCorrel, SERVICE=FAX, NUMBER=66133831
TXT:
Thanks for your interest in the Kofax solution.
Enjoy one of the last adventures ... connect to Kofax!

Bye,
Klaus
:ATT: NAME=1, APPLICATION=TOM_21902.doc
```

The "C1=" and "C2=" parameters will be returned in the "NFINFO:" section of the notification:

```
SUBJECT="Del: TC Open Message Format Description", TYPE=NOTIF, MCORR=00001639982
FROM: ACTIVE=YES, SERVICE=FAX, NUMBER=66133831
TO: ACTIVE=YES, SERVICE=MQ, NUMBER=MeOnMQSeries
NFINFO: STATUS=DEL, TIME=980303:163100, MCORR=00001639982, C1=4711,
C2=MyFirstMessageWithCorrel,
TXT:
TOPCALL Delivery Notification
-----

Message 00001639982 successfully sent to FAX,66133831
Time sent: 98-03-03 16:31:00
Subject: TC Open Message Format Description
Cost: 5 for GUEST
```

Note:

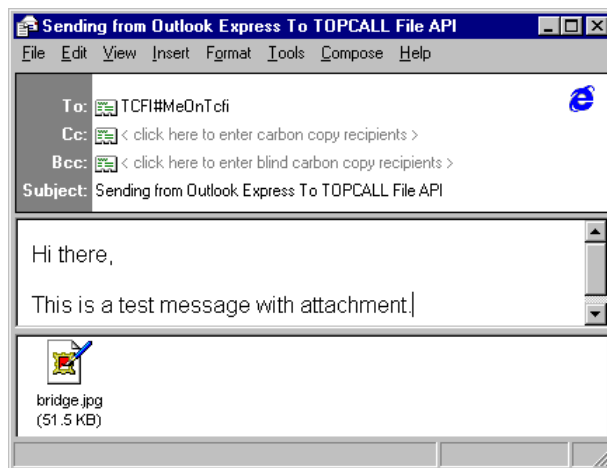
- The <NFINFO:> section gives information about the original message (when it was sent, what was the filename on KCS, and some more optional parameters), while the <FROM:> and <TO:> section are valid for the notification only.
- You can use C1 to C5 for storing correlation information (You may store numerical or textual information)
- If the correlation information contains commas or blanks, you must put it to double quotation marks.
- The <TXT:> section is build from the configured notification cover. See the *KCS Clients Administrator's Manual* on details how to build notification covers if the default cover installed does not fit your needs!

2.9 Receiving Messages from KCS

As we already learned in 2.4 “Defining the Originator”, the TC File Interface can be accessed by using the service “TCFI”; the MQSeries interface can be accessed by service “MQ”

So, on any connected mail system, you can address to Service “TCFI” / “MQ” ... and the message will be put to the TC_TO_FI directory / TC2.QUEUE !

Example file (sent from Internet mail client):



In the TC_TO_FI directory, you will get two files: A text file (“TCFIxxxx.MSG”)

```
SUBJECT=Sending from Outlook Express To TC File API, TYPE=NORM, P=HIGH, PCC=NORM, NF=NEG,
ARCHIVE=ALL, RESOLUTION=NORM, SCOPY=FIRST, HLINE=YES, CCTR=0, REMMSG=NO, MCCR=00001640022
FROM: ACTIVE=YES, SERVICE=SMTP, NUMBER=AI@kofax.com, ACTIVE=YES
TO: ACTIVE=YES, P=HIGH, NF=NO, ARCHIVE=ALL, RESOLUTION=HIGH, SCOPY=YES, HLINE=YES, CCTR=0,
REMMSG=NO, C1=00190658, C2=00000130, SERVICE=TCFI, NUMBER=MeOnTcfi, ACTIVE=YES
TXT:
Hi there,

This is a test message with attachment.
:ATT: NAME=C:\TCFI\TC_TO_FI\TCFI000A.AT, APPLICATION=bridge.jpg, COMMENT=image/jpeg
```

... and a binary attachment file (“TCFI000A.AT”), containing a JPEG image.

You see a lot of parameters here, like “ARCHIVE”, “RESOLUTION”. Again the general rule: Take what you need, ignore the rest!

When you receive the equivalent message in MQSeries, it will have two differences:

- There is no filename like “TCFIxxx.MSG”; instead, the CorrelId of the transaction message ends with four zero-bytes, indicating a transaction message.
- the “NAME=” parameter in the :ATT: section will indicate the CorrelId number of the attachment instead of the physical filename.

Receiving messages works also for incoming faxes that are routed via a distributor. For automatic fax distribution, you need to take a look at 3.5 “Directory Synchronization” to learn about the configuration.

2.10 Optional: Sending Notifications to KCS

Usually, it is not necessary to send any notifications back to KCS. As soon as the messages are successfully written to the corresponding file interface directory / MQSeries queue, KCS assumes that they are delivered. It is the responsibility of your application to handle any unexpected delivery failures.

However, if you want to establish real “end-to-end” notifications, you can!

The TC Open Message Format can be configured to request Notifications. This is indicated by a “NF=ALL” parameter in the message from KCS.

Example:

```
SUBJECT=Requesting a Notification from MQSeries, TYPE=NORM, P=HIGH, PCC=NORM, ARCHIVE=ALL,
RESOLUTION=NORM, SCOPY=FIRST, HLINE=YES, CCTR=0, REMMSG=NO, MCCORR=00001640022
FROM: ACTIVE=YES, SERVICE=SMTP, NUMBER=Ai@kofax.com, ACTIVE=YES
TO: ACTIVE=YES, NF=ALL, C1=00190658, C2=00000130, SERVICE=MQ, NUMBER=YouOnMQ, ACTIVE=YES
TXT:
Hi there,

I need a notif as soon as this is delivered!
```

If you find a “NF=ALL” parameter in the recipient section, you must provide a notification to KCS (in the FI_TO_TC directory / the 2TC.QUEUE) as soon as the message has been delivered (or, as soon as delivery has failed). This notification looks like

```
TYPE=NOTIF
FROM: SERVICE=MQ, NUMBER= YouOnMQ
TO: SERVICE=SMTP, NUMBER=Ai@kofax.com
NFINFO: C1=00190658, C2=00000130, STATUS=DEL
```

- The parameter TYPE must be set to NOTIF.
- The notification originator (FROM:) gives the recipient of the message.
- The notifications recipient (TO:) must specify the sender of the original message.
- In the section NFINFO: you must define all correlation information (usually C1 and C2) of the received message.
- The parameter STATUS specifies whether the message is a delivery notification (DEL) or a non-delivery notification (NONDEL).

Note:

If KCS requests a notification, you **must** return it! Otherwise, the KCS server will be filled with active send orders, and will stop operation after some time!

3. ADVANCED HINTS

As you already got your application working together with KCS, you may be interested in some special features that may be useful.

3.1 Supported Code Pages

The TC Open Message Format can be configured to use any code page supported by Windows. If you need code pages different from 437, 850, 852, 932, 1250 or 1252, you can also customize conversion tables (see the TCLINK manual for details).

If one of the characters used does not match perfectly to the code page used on the KCS server, a similar one replaces it.

3.2 Addressing To X.400 Address

In the previous section, we always used combinations of SERVICE and NUMBER to specify addresses. For X.400, this is a little different: There is not a single number, but many X.400 attributes:

```
TO: SERVICE=X400, C=at, A=umi-at, S=autoanswer  
TXT:  
This is a test to X.400.
```

See 4.5 “Sender/Recipient Address” for a complete list of available X.400 attributes!

3.3 Addressing To KCS Address Book Entries

KCS offers an own address book. This has some outstanding advantages:

- First of all: addressing is shorter (especially for X.400 addresses)
- Easier Maintenance: If e.g. a business partner changes the fax number, you only need to change it in a single position.
- Alternative numbers: You can specify more than one address for an address book entry. If e.g. the fax number is not responding, then the message can be sent to the alternative e-mail address.
- Additional parameters: You can specify many additional parameters (e.g. postal address, company name, salutation) for the address book entry. These parameters will be inserted to a coversheet (configurable by KCS system administrator).

Many advantages – but easy to use:

The address book is a KCS local service, therefore addressed by “SERVICE=TOPCALL”. But, in difference to usual addresses, there is no NUMBER, but a reference to the address book (called “shortname” in the KCS vocabulary; that’s where the SNAME= parameter comes from).

Example:

```
TO: SNAME=Microsoft Support, SERVICE=TOPCALL  
TXT:  
This is a simple test to the address book entry “Microsoft Support”.
```

3.4 Using CR/LF as Parameter Delimiter

All examples from section 2 “Getting Started” used comma-delimited parameters. However, you can also limit your parameters by CR, LF or CR/LF; so, the message

```
FROM: SERVICE=TCFI, NUMBER=MeOnTCFI
TO: SERVICE=SMTP, NUMBER=AI@kofax.com
TXT:
Third test:

Now I add an originator address!
```

Is identical to

```
FROM:
SERVICE=TCFI
NUMBER=MeOnTCFI
TO:
SERVICE=SMTP
NUMBER=AI@tofax.com
TXT:
Third test:

Now I add an originator address!
```

Note:

- If there are commas in one of your parameters, you still need to put the whole parameter to double quotation marks!

3.5 Directory Synchronization

We already mentioned Directory Synchronization at two points above:

- The DIRSYNC directory / DIRSYNC.QUEUE (that's where the Dirsync messages must be put in)
- The automatic distribution of inbound faxes.

The first is rather clear: we need a place to put the Dirsync messages.

The second is a little bit more complex; let me explain:

- If a fax is sent from somewhere in the world to the KCS fax interface, it usually has only a single external number (“66133” in our example). But via this number, a lot of users (thousands!) may be addressed – so there is the need for an automatic distribution.
- This is done by using DID: the sender dials some additional digits (so, the complete fax number is e.g. “66133831”), and the additional digits (“831”) are used to find out the real destination address.
- The information about the real destination address must be stored somewhere; that is what we call “KCS shadow users”. A KCS shadow user is something like an intermediate mailbox that is used to route inbound messages to their final destination.

And that is what we do via Dirsync: we create and maintain shadow users in order to get inbound faxes delivered to the final user.

There are two general possibilities to execute a Dirsync operation: The data can be put directly to the TXT-Section of the transaction file, or via an attached CSV-file.

3.5.1 Dirsync within the Transaction File

Example Dirsync message:

```

TYPE=DIRSYNC
TO: SERVICE=TCFI, NUMBER=void
TXT:
A,Name=AlonTCFI,Type=TYPE1,DID=810
M,Name=BMonTCFI,Type=TYPE1,DID=812
D,Name=XYZ

```

- The “TYPE=DIRSYNC” tells KCS that this is a Dirsync message.
- The “TO:” section must hold any valid recipient for compatibility to the TOM syntax; but this is ignored later on, the dirsinc message is never delivered to any user!
- The text part holds a number of Dirsync strings.

Dirsync strings consist of comma-delimited parameters (**Note:** No blanks allowed between the commas and the parameters!):

- The first parameter identifies the action to be taken (“A” for add user, “M” for modify, “D” for delete)
- The “Name=” parameter specifies the user short name (a reference that must be unique on every KCS system; similar to address book short names mentioned above)
- The “Type=” parameter indicated a shadow user template (all default parameters are taken from there; ask your local KCS administrator to create proper templates!)
- The “DID=” finally specifies the desired fax extension for the shadow user.
- There can be any number of additional parameters to fill additional fields of the shadow users; only limit is the maximum line length of 1023 characters.

This DirSync Message

- Creates the user “AlonTCFI” from the dirsinc template TYPE1, and enters a DID of 810
- Modifies the user “BMonTCFI” to have a DID of 812 (the user is created if not yet existing)
- Deletes the user “XYZ”

If a fax is sent to KCS with a number of “66133810”, the following will happen:

- KCS extracts the “810” from the number for use as a DID.
- KCS searches the shadow users for matching DIDs; it finds the “AlonTCFI”.
- The incoming fax is delivered to the shadow user “AlonTCFI”.
- As “AlonTCFI” is only a shadow user, the fax is forwarded to “SERVICE=TCFI, NUMBER=AlonTCFI”.
- You will find the fax in the TC_TO_FI directory some seconds after reception.

Hints/Restrictions:

- You can use the TC Open Message Format to update the KCS user store. It needs one string per user to add/modify/delete. For a detailed description of these strings, see the TC/LINK Manual.
- Every Dirsync string must completely be written on a single line! This limits the maximum length to 1023 characters (including the CR/LF pair).
- Make sure that the given Dirsync template matches the parameter order of the Dirsync strings!
- If you use an address book entry as a template, you can also change the address book by using Dirsync!

3.5.2 Dirsync with an Attached CSV File

Dirsync within the transaction file uses the general syntax “ParameterName=Value,” although for the first parameters the parameter-names are not obligatory. The message type “DIRSYNC_CSV” is able to use an attached CSV-file for directory synchronization. The first row of the CSV-file consists of the parameter-names, the rest of the rows are filled with values. Parameters/columns that have always the same value don’t necessarily need an own column. It is possible to define them by quotes as constants.

In the parameter-section CSVINFO the following parameters allow configuring the column-names and values of the CSV-file:

Parameter	Default	Description
MODE	MODE	Name of column which values define if users/recipients shall be added, modified or deleted
NAME	NAME	Name of column in that the KCS user-ids are specified

TYPE	TYPE	Name of column in that the KCS user- or recipient-templates (shadow-users) are specified, defining the mapping of the rest of the attributes to be synchronized
ADD	A	Value in Mode-column for adding
MODIFY	M	Value in Mode-column for modifying
DELETE	D	Value in Mode-column for deleting
LINKTYPE	"FI" or "MQ"	Defines the Link-type the added user or recipient belongs to
DELIMITER	,	Defines the column-delimiter in the CSV-file

If the values are quoted, they are interpreted as constants. That means, that no column of that name is searched for in the CSV-file, but the quoted value itself is taken for the whole synchronization-process. That is only possible for MODE, TYPE and LINKTYPE:

- *MODE=xyz* would refer to a column that specifies the operation to be executed (default: either A, M, or D; or one of the values defined the same way with the ADD, MODIFY and DELETE parameters). If *MODE="A"*, the quotes would make the value a constant. That means no column with the name A is searched for, but the value A itself is used to define the operation. Only "A", "M", or "D" are valid constants for the MODE parameter.
- *TYPE=template* requests the CSV-file to have a column named template. This column has to provide the names of users or recipients on the KCS server for mapping the rest of the attributes. *TYPE="template"* would not search for a column of the name template, but would search for a user or recipient of the name template itself on the KCS server
- *LINKTYPE="MQ"* (or "FI") is the default for the LINKTYPE parameter. The LINKTYPE specifies to which Link a user belongs (that means, which Link is allowed to update him). As CSV-files may be produced by mail-systems like MS-Exchange, users may not only belong to the Link by which they are updated, but also to other Link-types, so LINKTYPE is configurable. *LINKTYPE="MX"* would update MS-Exchange users on KCS. If the value is quoted like the example, no column in the CSV-file is required.

The rest of the columns of the CSV-file are mapped through the shadow-user templates defined in the TYPE-column. The following screenshot shows a KCS user of the name ms_template used for dirsinc. The values between the \$-signs have to be the names of the CSV-columns.

Example for transaction-file (The CSV-file is attached in the ATT-section):

```
TYPE=DIRSYNC_CSV
TO: SERVICE=TCMQ, NUMBER=void
CSVINFO: MODE=mode, NAME=DirectoryName, TYPE=template
:ATT: NAME=1, APPLICATION= update.csv
```

Example for CSV-file

mode,DirectoryName,template,Company,Department,Fullname,Freetext
A,ms_csvdirsnc1,ms_template,comp data,dep data,fullname data, freetext data
M,ms_csvdirsnc2,recipient_template,data1,data2,data3,data4
D,ms_csvdirsnc3

The first data-line of the CSV-file creates the user `ms_csvdirsinc1` (using the template-user shown above). The values for Company, Department, Full Name and Freetext are taken from the rest of the CSV-line. The User-belongs-to field is updated according to the default Linktype MQ (in this example the Dirsync was executed by Link-MQ). The service-name on that server for MQ is called "IBM MQSeries".

For all parameters values are defined in the registry, which are used if nothing is specified in the transaction-file. If no registry-values exist, the keys are created with the listed default values.

Registry Keys (...\\TOPCALL\\TCLINKxy\\Dirsync_CSV):

Registry Key	Type	Default	Description
ColumnMode	STRING	MODE	Default name for column Mode
ColumnName	STRING	NAME	Default name for column Name
ColumnType	STRING	TYPE	Default name for column Type
OperationAdd	STRING	A	Value in Mode-column for adding
OperationModify	STRING	M	Value in Mode-column for modifying
OperationDelete	STRING	D	Value in Mode-column for deleting
LinkType	STRING	"xy"	Default Link-type the added user belongs to
Delimiter	DWORD	44 (,)	Defines the column-delimiter in the CSV-file as ASCII-value

xy stands for the two letters specifying the Linktype using the TC Open Message Format.

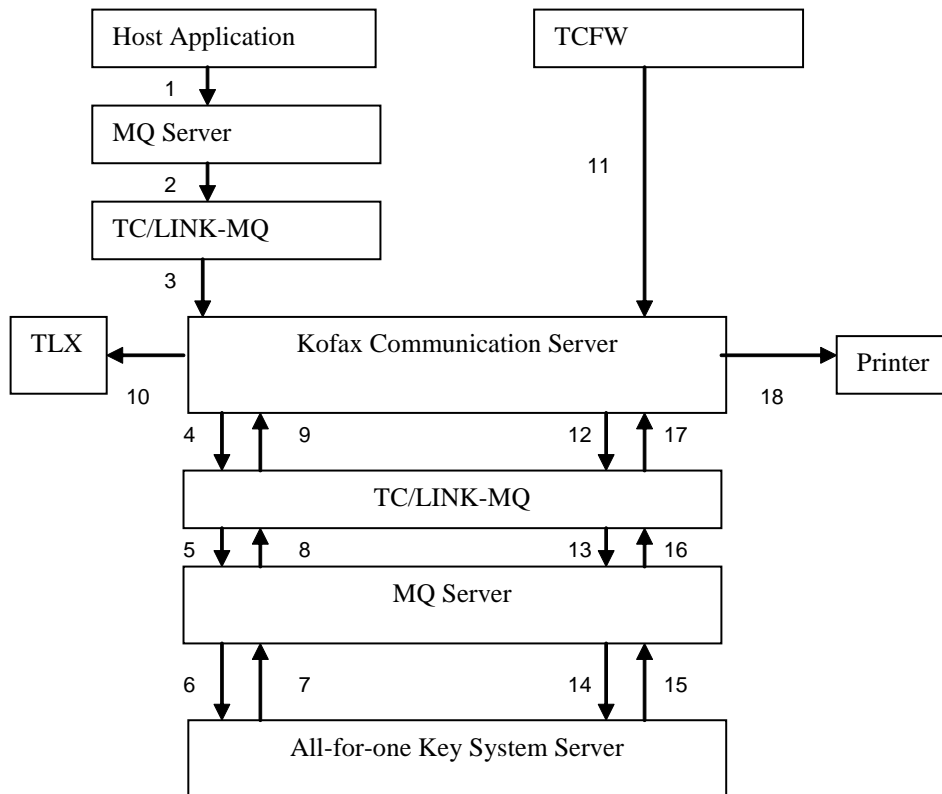
For a detailed description of how Directory Synchronization generally works, and especially how to use shadow user templates, see the TC/LINK Manual.

3.6 All-for-one Key System Support

TC/LINK-FI and MQ are integrated with the All-For-One Banking Key System. This means that the Key System can be used for validation of outgoing and incoming messages.

The following examples and explanations refer to TC/LINK-MQ; nevertheless it works the same way with TC/LINK-FI. Instead of the MQ-Server there would be the Transfer Directories.

The following picture illustrates the typical message flow for calculation or validation of Banking Keys. For better understanding, this message flow contains neither error handling nor delivery notifications.



Authorizing outgoing telex:

- 1) A host application posts a TOM message to an MQ Server queue serviced by TC/LINK-MQ. The message contains banking key information, and is sent to a telex number. The Key System is specified as an additional AUTH recipient.
- 2) TC/LINK-MQ fetches the message from the MQ Server.
- 3) TC/LINK-MQ makes the telex recipient inactive and posts the message to KCS.
- 4) As the AUTH recipient had an MQ address, it is fetched by another instance of TC/LINK-MQ and
- 5) Posted to a queue on another MQ Server.
- 6) The All-for-one application fetches the message from the queue and calculates the key.
- 7) It creates a special TOM notification, containing the calculated key, a status, the KCS message id and modified message text, and posts this notification to a queue on the MQ server.
- 8) TC/LINK-MQ retrieves the notification from the queue.
- 9) If the notification status signals success, TC/LINK-MQ authorizes the original message on KCS: the AUTH recipient is set inactive, and the telex recipient is set active instead. Additionally, the modified banking key information and message text are stored in the original message.
- 10) KCS now sends the message to it's real destination: the telex number

Verifying banking keys on incoming telex:

- 11) The person responsible for distribution of incoming telex message creates a new message via TCFW. This message contains the banking key information and text from the received telex. It is sent to a printer (real recipient) and to the Key System (as an Authorizer). TCFW makes the real recipient inactive and routes the message to the Key System.
- 12) As the AUTH recipient had an MQ address, it is fetched by another instance of TC/LINK-MQ and
- 13) Posted to a queue on another MQ Server.
- 14) The All-for-one application fetches the message from the queue and validates the key.
- 15) It creates a special TOM notification, containing the validation result, the KCS message id and modified message text, and posts this notification to a queue on the MQ server.
- 16) TC/LINK-MQ retrieves the notification from the queue.
- 17) If the notification status signals success, TC/LINK-MQ authorizes the original message on KCS: the AUTH recipient is set inactive, and the original recipient is set active instead. Additionally, the modified banking key information and message text are stored in the original message.
- 18) KCS now sends the message to it's real destination: the printer

3.6.1.1 Keywords Defined for Banking Key Information

The transaction file object "KEYINFO" holds the Banking Key Information. It must be placed before the message text. For example:

KEYINFO:

S_ADDRESSCODE=....

.....

TEXT:

.....

Possible parameters for the KEYINFO object:

Parameter	Value Range, Format	Default	Meaning	Comment
S_Date	0-9 String	-	Date	Value range checked by Key System.
S_TYpe	CHECK / BUILD	-	Existing key will be checked or new key will be calculated.	
S_CURRENCY1 – S_CURRENCY100	A-Z,a-z String	-	Currency, e.g. ATS. The keys must start with 1 without any missing numbers.	Value range checked by Key System
S_AMOUNT1 – S_AMOUNT100	0-9 Integer	-	Amount. The keys must start with 1 without any missing numbers.	Value range checked by Key System
S_REFeRence	0-9,A-Z,a-z String	-	Reference	Value range checked by Key System
S_ADdresscode	Free String	-	Address code	
S_KEY	Free String	-	Either the key calculated or the key to check	
S_LEtter	A-Z,a-z String	-	Part of key	Value range checked by Key System
S_TOTalamount	0-9 Integer	-	Total amount	Value range checked by Key System.
S_BITpattern	Free String	-	Pattern for error recognition	
S_NUMberseries	0-9 Integer	-	Number series	Value range checked by Key System
S_KEYERror	Free String	-	Error description	

For messages to KCS, short forms are accepted for certain parameters. These short forms are indicated by the **BOLD UPPERCASE** letters in the syntax table above (e.g. "S_Date" means that you can use either "S_D=" or "S_DATE=")

Note:

IBM MQ-Series must be configured properly to convert from EBCDIC to ASCII and ASCII to EBCDIC. All characters used must have a representation in the code page used on the link computer and in the used TCOSS code page. Otherwise the character conversions may lead to different representations.

For backward compatibility, TC/LINK-MQ only uses these keywords in messages to MQ if bit IF_LEVEL2 in registry value Options\InterfaceLevel.

3.6.1.2 Authorizing

If a message shall be checked by the All-for-one Key System, its list of recipients must hold the Key System as an Authorizer. With TC/LINK-MQ, you can use an AUTH object to specify the Authorizer. At the moment, an MQ (or FI) address must be used for the Key System.

When TC/LINK recognizes an Authorizer in a message, it sets all other recipients inactive and routes the message to the Authorizer (i.e. the All-for-one Key Server).

When the All-for-one Key System gets the message, it reads the banking key data and calculates or validates the key. In addition, it adds information to the message text.

The Key System returns a new type of notification to TC/LINK-MQ, including the message type AUTH, a NFINFO object with STATUS, C1 and C2, the modified KEYINFO object, and the modified message text.

For example:

```
TYPE=AUTH
NFINFO: STATUS=DEL, C1=00000017, C2 =00003024
KEYINFO: S_TYPE=CHECK, S_KEY=0129384756ABC, ....
TXT:
ZCZC
NYCSTBC
.ZRHHOST  CREDIT SUISSE ZURICH
....
Key (012984756ABC) ....
NNNN
```

Parameter STATUS in the NFINFO object indicates whether the calculation succeeded (DEL) or failed (NONDEL). Parameters C1 and C2 hold the KCS message id of the original message, taken from the corresponding parameters in the active AUTH object of the message sent to the Key System.

On receipt of a message of type AUTH, TC/LINK-MQ opens the original message in the KCS message store.

If parameter STATUS signals success (DEL), TC/LINK-MQ authorizes the original message: The currently active recipient (the Authorizer) is set inactive. If there is another authorizer that has not yet been serviced, TC/LINK-MQ sets him active. Otherwise, all other inactive recipients are set to active. Thus, the message will be routed either to the next authorizer (if there is any) or to the real recipients. The modified banking key data and the modified message text are written to the original message, before it is posted to KCS.

If parameter STATUS signals failure (NONDEL), TC/LINK-MQ creates a non-delivery notification for the original message. This non-delivery notification includes the modified banking key information and the modified message body.

3.6.1.3 Error Handling

Original Message Timeout:

If TC/LINK-MQ cannot open the original message, because it does not exist in the message store any more, it considers the message to be timed out and discards the notification. In this case the original message has already been negatively terminated by KCS.

Loss of connection to KCS:

If the connection to the KCS server gets lost during authorizing or creation of a non-delivery notification, the notification is backed out on MQ and TC/LINK enters idle mode.

When the idle timeout expires, TC/LINK-MQ will fetch the notification again from the queue.

Conversion Errors:

If the notification from the Key System for some reason cannot be converted, but contains at least the KCS message id of the original message, TC/LINK-MQ terminates the original message via a non-delivery notification.

Corrupted Message from Key System:

If the notification from the Key System does not contain the KCS message id of the original message, TC/LINK-MQ cannot correlate it to any existing KCS send order. It writes an event log entry and discards the notification.

3.7 Syntax Errors in Messages to KCS

If a message to KCS could not be sent due to a syntax error, a non-delivery notification consisting of two files is generated in the NOTIF directory / NOTIF.QUEUE:

- a non-delivery notification file with the name *TCFIxxxx.NTF* (or, on MQ, CorrelId ending with 0000) and
- an attachment *TCFIxxxx.AT* (or, on MQ, CorrelId ending with 0001) ...the original message

The following error messages can occur in the non-delivery file:

Error indication	Corrective Action
Error in %1 Part: Unknown parameter %2	Illegal parameter given (may also be a missing delimiter between parameters)
Error in %1 Part: Illegal value of parameter %2	Check out the possible values of the given parameter!
Missing message recipient	At least one recipient must be given for a message.
Unknown service	Make sure that the used services are defined on KCS!
Illegal parameters for the given service	The service can not have the given parameters (e.g. service "TOPCALL" can not have a "CN=" parameter)

Possible values of %1: "General", "Sender", "Recipient", "Correlation", ":", "ATT:", ":", "INC:".

Example of a rejected message:

```
TYPE=NOTIF,NF=NO
NFINFO: STATUS=NONDEL, TIME=000405:092330, LACTION=LN, LNOTE=Syntax error
TO: ACTIVE=YES, SNAME=, SERVICE=FREE, ACTIVE=YES, NUMBER=MS
TXT:
-----
Found an error in transaction file (line 1):
Error in General Part: illegal value of parameter NF ... NIX

Original transaction file is attached.
-----
:ATT: NAME=C:\TCIFI\NOTIF\TCFI000A.AT
```

The notification file (TCFI000A.NTF) contains a text line 'ERROR in general part' which means that there is a syntax error in the general part. The attachment (:ATT:) contains the original message:

```
SUBJECT="Test of TCFI", NF=NIX
TO: SERVICE=FAX, NUMBER=66133831
TXT:
This is a sample with a wrong syntax.
Test end.
```

Note: The message was rejected because “NIX” is an invalid value for the parameter “NF”.

4. SYNTAX REFERENCE

If you worked through the previous chapters, you are already a TC Open Message Format expert. For your reference, this chapter gives a complete listing of all parameters allowed, and some limits that must be kept.

The TC Open Message Format is a mapping of the KCS-internal envelope structure to an ASCII format. We have been careful to keep the format simple for simple messages.

Note:

- For messages to KCS, short forms are accepted for certain parameters. These short forms are indicated by the **BOLD UPPERCASE** letters in the syntax tables below (e.g. if you read "**su**bject" this means that you can use either "SU=" or "SUBJECT=")
- For messages from KCS, always the full format is generated ("SUBJECT=")
- Maximum line length is 1023 including CR/LF.
- KCS creates lines terminated with CR/LF and accepts CR, LF or CR/LF as line termination.
- The code page for both directions is the same. It is configurable during TC/LINK setup.

4.1 TC Open Message Format Structure Overview

The interface syntax is based on the KCS-internal form of the message as an object structure (TCSI objects). Each object may hold values and other objects.

Objects in the header are defined with a key word at the beginning of a line followed by a colon (":"). Objects in the content of the message are also prefixed with a colon to make them unique.

Values of objects are set with: parameter=value [,parameter=value ...] Values may include blanks; values with commas must be inside double quotes. Leading blanks before parameters and values are ignored. Leading blanks can be specified if the value is inside double quotation marks.

The transaction file must have a specific format:

General Message parameters part
Sender parameters part
Recipient(s) parameter part
Message part

Example:

SUBJECT="Test document"
FROM: SNAME=RF, SERVICE=INT, NUMBER=RF
TO: SERVICE=FAX, NUMBER=1234567
CC: SERVICE=FAX, NUMBER=7654321
This is a test.

It is important that you always use this order. You may not, for instance, define general message parameters after the sender parameters.

In the transaction file, all key words must be uppercase.

4.2 General Message Parameters

The first part of the header holds general parameters for the message as well as for the sender and recipient objects.

Parameter	Value Range, Format	Default	Meaning	Comment
SUBject	free string	from template	the subject of the message	
TEmpIate	[folder/]msgname / NO	see below	the KCS message to be taken as template	TE=NO: no template
TYpe	NORM / ROUTING / NOTIF / DIRSYNC / AUTH	NORM	the type of the message	Dirsync messages can be addressed to any valid recipient.
P	Low / Norm / High / SUPERIOR_1 – SUPERIOR_27	-	sending priority TO:, AUTH:	Sets default for TO recipients.
PCc	Low / Norm / High / SUPERIOR_1 – SUPERIOR_27	-	sending priority CC:, BCC:	Sets default for CC recipients.
NF	NO / NEG / POS / ALL	-	notification requested	sets def. for all recipients
ARchive	NO / NEG / POS / ALL	-	archive entries	sets def. for all recipients
Resolution	NORM / HIGH	NORM	the resolution for sending	sets def. for all recipients
SCopy	NO / ALL / FIRST	NO	sending copies	defines how sending copy is set for each of the recipients
HLine	NO / YES	-	FAX headerline	sets def. for all recipients
CCtr	free string	-	the cost center	sets def. for all recipients
Date	yyymmdd, or yyyyymmdd	-	required date of sending / reception date	sets def. for all recipients
Time	hh[mm[ss]]	-	required time of sending / reception date	sets def. for all recipients
EXTIME	hh[mm[ss]]	-	Latest delivery time (time of day) **	sets def. for all recipients
EXDate	yyymmdd, or yyyyymmdd	-	Latest delivery time (date) **	sets def. for all recipients
EXTIMERel	[h[h]h]mm	-	Latest delivery time (relative) **	sets def. For all recipients
REMsg	NO / YES	NO	remove message from KCS active Mail folder as soon as all entries are terminated	sets default for all recipients, used e.g., for sending with mailmerge printing
MCorr	free string	-	Message name	
NPAGE	Integer	-	Page count of message from KCS *	
DOCERR	2 digit error code (e.g. LV)	-	Reception error code *	
SUSPDUPL	0 / 1	-	Suspected duplication of message *	

* ... Fields are only used in direction from KCS, if bit IF_LEVEL1 in registry value "Options\InterfaceLevel" is set !

** ... Fields are only used in direction from KCS if bit IF_LEVEL2 in registry value "Options\InterfaceLevel" is set !

The KCS Message Template (TE=):

Only for the direction to KCS. The template defines:

- General message options
- The cover sheet and forms
- Standard recipientS (e.g., mandatory authorizer).

If this is omitted, there are the following fallbacks:

- From the registry (key "...Options\DefaultTemplate")
- From the originator's KCS user profile (if shadow user exists)
- From the configured GUEST user profile.

The default template can be switched off with TE=NO.

DATE/TIME Issues:

- TIME and EXTIME are required to have at least two digits. All missing digits will default to zero. So, "TIME=19" or "TIME=1900" will be equivalent to the time given above.
- DATE and EXDATE can also be in four-digit-year format; so, "EXDATE=19990923" is equivalent to "EXDATE=990923"
- If the two-year format is used in messages to KCS, years 00 to 92 are interpreted as 2000 to 2092.
- In messages from KCS, a two-digit year format is used. Values from 00 to 92 must be interpreted as 2000 to 2092.
- To set latest delivery to INFINITE, KCS uses the value EXDATE=610119, EXTIME=031407 (i.e. January 19, 2061).
- EXTIME is required to have at least two digits. All missing digits will default to zero. So, "EXTIME=19" is equivalent to "EXTIME=1900".
- EXTIMEREL is required to have at least two digits. All missing digits will default to zero. So, "EXTIMEREL=45" is equivalent to "EXTIMEREL=00045".
- Either use EXTIME and EXDATE or use EXTIMEREL for an object. Mixing absolute and relative expiration times is not allowed.
- The time specified is a KCS time (based on the time zone of the KCS server)

4.2.1 Default Handling of Sending Parameters

Parameter values for the recipients are taken from the following objects in this order:

1. From ++directives in message text or attachments (see TCLINK manual)
2. As set in the recipient/sender object
3. As set in the "General" part of the transaction file
4. From the KCS template of the sender
5. The "Guest" user configured with TCLINK
6. From TCLINK configuration (registry)
7. Default values defined on KCS

4.3 The Sender Object (FROM:)

The originator is required if notifications are to be routed back to the sender. It may be omitted if a default originator is configured at TC/LINK-FI (TC/LINK-MQ).

If there is no FROM: field in the transaction file and the default originator settings (registry keys DefaultOriginatorNumber, DefaultOriginatorService) do not refer to an existing KCS user, then TCLINK uses the Guest User's send options and permissions. This can lead to problems if the registry key CheckRights is set to 1. (See also the general TC/LINK manual.)

Parameter	Value range, format	Default	Meaning	Comment
Active	YES / NO	YES	Address-active flag	
SName	Free string	-	Short name	
COmpany	Free string	-	Company	
Dept	Free string	-	Department	
Name	Free string	-	Recipient name	
Salut	Free string	-	Salutation	
TXt	Free string	-	Free text	
C1...C6	Free string	-	Correlation information fields	

4.4 The Recipient Objects (TO, CC, BCC, AUTH)

It is possible to send one message to multiple recipients. For each recipient one line is required, starting with one of the recipient tokens (TO:, CC:, BCC: or AUTH:). At least one active **TO:** recipient per message is required.

All sender parameters, plus the following can be used:

Parameter	Value Range, Format	Default	Meaning	Comment
P	Low / Norm / High / SUPERIOR_1 – SUPERIOR_27	-	Priority	
NF	NO / NEG / POS / ALL	NEG	notification requested	
AR chive	NO / NEG / POS / ALL	ALL	archive entries	
Res olution	NORM / HIGH	-	the resolution for sending	
SC opy	NO / ALL / FIRST	NO	sending copy	defines how sending copy is set for each of the recipients
H line	NO / YES	-	FAX headerline	can be used to turn off headerline for internal sending
CC tr	Free string	-	the cost center	
D ate	yymmdd or yyyyymmdd	-	required date of sending	
T ime	hh[mm[ss]]	-	required time of sending	Missing digits default to 0
EX TIME	hh[mm[ss]]	-	Latest delivery time (time of day)	Missing digits default to 0 If only EXDATE is specified, EXTIME defaults to 000000
EX Date	yymmdd or yyyyymmdd	-	Latest delivery time (date)	If only EXTIME is specified, EXDATE defaults to the current date (today)
EX TIMERel	[h[h[h]]]mm	-	Latest delivery time (relative)	100 = 60 minutes Missing digits for hours default to 0
RE Mmsg	NO / YES	-	remove message from KCS active mail folder as soon as possible	(keep entries for cyclical erasure)
C1 .. C5	Free string	-	Correlation Information field 1 .. 5	may be freely filled, values are returned in notification; the maximum length is 127 characters

4.5 Sender/Recipient Addressing

4.5.1 Standard Format for Sender/Recipient Address

A recipient/sender may have several alternative addresses. Alternative addresses belong to one recipient and are therefore defined all in one line. For example:

TO: SERVICE=FAX, NUMBER=999, SERVICE=FAX, NUMBER=151

That way KCS tries first to send to number 999, and if that does not succeed it tries to send to the second number 151. It is also possible to send to alternative services.

Multiple recipients are defined by multiple recipient lines, starting with one of the four recipient tokens (TO:, CC:, BCC:, AUTH:). Multiple recipients' means each of the recipients get the message, regardless if the other send-orders are processed successfully.

At least one active address per active recipient is required. An address starts with the parameter SERVICE, and then the fields of the address follow. The required fields depend on the address type.

Parameter	Value Range, Format	Default	Meaning	For Address type (as defined in Service)
S ervice	free string	Configured at TC/LINK	Service	all
A Ctive	YES / NO	YES	Address-active flag	all
N umber	free string	-	Number	FAX, TX, TTX, FREE, TOPCALL
A B	free string	Empty	Answerback	FAX, TX, TTX
N Ode	free string	Empty	KCS Node	TOPCALL
C	free string	Empty	Country	X.400
A	free string	Empty	Admin. Mgmt domain	X.400
P	free string	Empty	Private domain	X.400
OU1 .. OU4	free string	Empty	Organizational Unit 1 .. 4	X.400
O	free string	Empty	Organization	X.400
G	free string	Empty	Given Name	X.400
S	free string	Mandatory	Surname	X.400
I	free string	Empty	Initials	X.400
Q	free string	Empty	Generation Qualifier	X.400
CN	free string	Empty	Common Name	X.400; not supported on KCS

For easy input, TC Open Message Format accepts addresses that start with the NUMBER parameter - in this case the default recipient service (defined at TC/LINK setup) is used. This requires that if SERVICE and NUMBER are specified within one address, NUMBER must directly follow SERVICE.

4.5.2 Shortcut Format for Recipient Address

For easy input, the TC Open Message Format accepts a short form for a recipient: The first four parameters may be specified as comma-delimited values (without the corresponding 'parameter=').

Format:

```
TO: [fullname,][service,]number[,answerback][,parameter=value ...]
```

Note:

- Items are checked from left to right. So if you specify 2 Items, but no service, they will be interpreted as full name and number, not number and answerback.
- The service can be omitted; in this case, the default service as defined on TC/LINK will be used.
- The full name is optional too; there is no default.
- X.400 addresses can not be used with shortcut addressing; they must always be entered using the full syntax (e.g. <TO: SE=X400, G=AI, A=umi-at, C=at>!

Example short form:

```
TO: Maxi, 6613321, P=H, N=4711, SERV=TELEX, N=111388
```

Equivalent normal form:

```
TO: NAME=Maxi, Priority=HIGH
SERV=FAX, N=6613321      (configured default recipient service = FAX)
SERV=FAX, N=4711
SERV=TELEX, N=111388
```

Note:

- Short-form addressing is not allowed for the originator address!
- Every short form address must start with "TO:" and consist of only one line.
- Note that this is an example for alternative addressing that means only one of the defined addresses get the message!

4.6 Banking Key Information (KEYINFO:)

This optional message part is only needed for integration with the All-for-One Banking Key Server.

Parameter	Value Range, Format	Default	Meaning	Comment
S_Date	0-9 String	-	Date	Value range checked by Key System.
S_Type	CHECK / BUILD	-	Existing key will be checked or new key will be calculated.	
S_CURRENCY1 – S_CURRENCY100	A-Z,a-z String	-	Currency, e.g. ATS. The keys must start with 1 without any missing numbers.	Value range checked by Key System
S_AMOUNT1 – S_AMOUNT100	0-9 Integer	-	Amount. The keys must start with 1 without any missing numbers.	Value range checked by Key System
S_Referenc	0-9,A-Z,a-z String	-	Reference	Value range checked by Key System
S_Addresscode	Free String	-	Address code	
S_KEY	Free String	-	Either the key calculated or the key to check	
S_Letter	A-Z,a-z String	-	Part of key	Value range checked by Key System
S_Totamount	0-9 Integer	-	Total amount	Value range checked by Key System.
S_BITpattern	Free String	-	Pattern for error recognition	
S_NUMberseries	0-9 Integer	-	Number series	Value range checked by Key System
S_KEYError	Free String	-	Error description	

Note:

In messages from KCS, this object is only created if bit IF_LEVEL2 in registry value "Options\InterfaceLevel" is set.

All characters used must have a representation in the code page used on the link computer and in the used TCOSS code page. Otherwise the character conversions may lead to different representations.

4.7 The Correlation Object (NFINFO:)

This exists only in messages of TYPE=NOTIF. A notification always applies to a certain entry of a certain message. This notification object contains the message and entry correlation information (for message and entry) plus the send status of the original entry.

Parameter	Value Range, Format	Default	Meaning	Comment
STatus	NONDEL / DEL	-	status of referenced entry	
Time	yymmdd:hhmmss	-	time of delivery or negative termination	
DOcnr	string: 'ddd-ppp'	-	the document number and the last confirmed page sent	as in headerline or cover of sent message
LAction	string	-	last action of delivery agent	2 character error code as defined on KCS
LNote	string	-	last note of delivery agent	error description as defined on KCS
COST	integer	-	the cost	unit of cost as defined for the installation
MCorr	string, max 11 characters	-	message correlation info	as specified for original message
C1 .. C5	free string	-	Correlation Information field 1 .. 5	as specified for original Recipient

- **Original message to KCS:** The TC Open Message Format notification returns all correlation fields (if these fields are specified in the original message and entry).
- **Original message from KCS:** When a notification is requested for a recipient (NOTIF=YES), TC Open Message Format specifies C1 and C2 as correlation information for this recipient. C1 and C2 are 32-bit integers in decimal form. Only STATUS, C1 and C2 need to be returned by the application in the NFINFO: of the notification. The notification should be addressed to the originator of the message. All other parameters are optional; DOCNR and MCORR are ignored.

4.8 CSV File Information for Dirsync (CSVINFO:)

The message type "DIRSYNC_CSV" is able to use an attached CSV-file for directory synchronization. The first row of the CSV-file consists of the parameter-names, the rest of the rows are filled with values. Parameters/columns that have always the same value don't necessarily need an own column. It is possible to define them by quotes as constants.

In the parameter-section CSVINFO: the following parameters allow configuring the column-names and values of the CSV-file:

Parameter	Default	Description
MODE	MODE	Name of column which values define if users/recipients shall be added, modified or deleted
NAME	NAME	Name of column in that the KCS user-ids are specified
TYPE	TYPE	Name of column in that the KCS user- or recipient-templates (shadow-users) are specified, defining the mapping of the rest of the attributes to be synchronized
ADD	A	Value in Mode-column for adding
MODIFY	M	Value in Mode-column for modifying
DELETE	D	Value in Mode-column for deleting
LINKTYPE	"FI" or "MQ"	Defines the Link-type the added user or recipient belongs to

4.9 The Start of Message Content (TXT:)

This tag defines the beginning of the message content.

If **TXT:** is omitted (in the direction to KCS), the first line without valid keyword (object: or parameter=) will be used as the beginning of the message content. The content will be taken as TCI text, which is a superset of plain text. Form feeds are interpreted as page breaks.

Text Format Conversions:

TC/LINK does no format conversion. It delivers text messages "as they are" (according to the PC and TC codepage setup!).

4.10 Attached Files (:ATT:)

Attached files are inserted in the KCS message as attached (embedded) objects. You can select the export format for image blocks. The TC Open Message Format exports all attached binary and image objects as attached files (exception: when TCI is selected as output format, image blocks are encoded in text).

Parameter	Value Range, Format	Default	Meaning	Comment
NAme	String	mandatory	Full DOS filespec	physical file name or MQ CorrelId
APplication	String	-	File name with extension, defining the file type	logical file name
COMment	free string	-	Comment	
ATTSize	Integer	-	File size of attachment *	

* ... Fields are only used in direction from KCS, if bit IF_LEVEL1 in registry key "Options\InterfaceLevel" is set!

4.11 Attached KCS Objects (:INC:)

This tag includes a KCS message as linked object. The KCS message may be a full message or it may hold just an overlay (a form).

Parameter	Value Range, Format	Default	Meaning	Comment
NAme	String	Mandatory	KCS [folder/]file name	Folder default is taken from registry
COMment	free string	-	Comment	

Example:

```
:INC: NA=FIS/PRICES, COM=public price list
```

This command will insert the document "FIS/PRICES" to the outgoing message.

4.12 Authorizing Notifications (from All-for-One server)

The All-for-One banking key server returns a special type of notification to KCS, these notifications are used to authorize (or terminate) a send order on the KCS server.

- The parameter TYPE must be set to AUTH.
- There must be a NFINFO entry with (at least) parameters C1, C2 and STATUS.
- Depending on the STATUS, the KCS send order specified by C1 and C2 is authorized (if STATUS is DEL) or terminated negatively (if STATUS is NONDEL).
- They must be text-only (no attachments).
- If there is a KEYINFO object, it is copied to the original message (or non-delivery notification).
- The notification text will be copied to the original message (or non-delivery notification).

4.13 Important Notes on TC Open Message Format Syntax

To avoid problems when working with the TC Open Message Format, pay attention to a few more points:

- All transaction files to KCS must have the specified order of parameters (general - originator - recipient – message)!
- All files / MQSeries messages will be removed automatically after transmission to KCS.
- All keywords must be written in CAPITAL LETTERS.
- Mixing short form and normal form addressing is not allowed (e.g., <TO: NA=Mr. Test, FAX, 66133831> is not valid).
- Blank lines within the parameter section are not allowed. Blank lines cause TC/LINK to ignore all parameters that follow a blank line (because TC/LINK interprets a blank line as start of the text part!)
- When you use the TC Open Message Format without the key word TXT: make sure that the text does not start with any key word.
- :INC: and :ATT: are ONLY interpreted when found in the text body of a message, NOT in any text, binary, or image attachments.

5. APPENDIX

5.1 Registry Values Used By TC Open Message Format

Some parameters of the TC Open Message Format are configurable in the Windows registry.

HKLM\Software\TOPCALL\TCLINKxx\Options

Registry Key	Type	Default	Possible Values / Meaning
CutBlanksInText	DWORD	0	If 1, all trailing blanks in message text lines are removed. Useful if the message host sends lines with a fixed length, padded with blanks. (Incoming only)
DefaultFolder	STRING	+MAIL5V	Defines the default TOS-folder. This is used for :INC: - objects with no defined folder. E.g. entering "FIS" will take all included objects from FIS folder by default.
DefaultOriginatorNumber	STRING	-	Defines the default number for the originator, if it is not defined in the FROM: object of the TOM message. Must be only the Number or - in case of X400 - the full X.400 address
DefaultOriginatorService	STRING	MQ	Defines the default service for the originator, if not defined in the FROM: object of the TOM message.
DefaultRecipientService	STRING	FAX	Defines the default service for the recipient, if it is not defined in the TO: object of the TOM message.
DefaultTemplate	STRING	-	Defines the default template, if there is no "TEMPLATE"-Parameter in the general part of the TOM-message.
DefaultUserName	STRING	-	Defines the default short name for the originator, if it is not defined in the FROM: object of the TOM message.
InterfaceLevel	DWORD	0	Bit field indicating the version of TOM Format to be used in messages from KCS. 0 ... No extensions. 1 (IF_LEVEL1)... use reception time, attachment size, page count 2 (IF_LEVEL2)... use latest delivery time, KEYINFO 3 ... use all possible extensions
MaxActiveRecp	DWORD (1..100)	1	Indicates the maximum number of active recipients per message from KCS.
NonDelEventLog	DWORD	0	When set to 1 for each immediate non-delivery notification an event log entry is created. Immediate non-delivery notifications are created by TC/LINK before a message comes to KCS; they are not counted by TC/Monitor.

HKLM\Software\TOPCALL\TCLINKxx\Dirsync_CSV

Registry Key	Type	Default	Description
ColumnMode	STRING	MODE	Default name for column Mode
ColumnName	STRING	NAME	Default name for column Name
ColumnType	STRING	TYPE	Default name for column Type
OperationAdd	STRING	A	Value in Mode-column for adding
OperationModify	STRING	M	Value in Mode-column for modifying
OperationDelete	STRING	D	Value in Mode-column for deleting
LinkType	STRING	"MQ"/"FI"	Default Link-type the added user belongs to
Delimiter	DWORD (0..255)	44 (,)	Defines the column-delimiter in the CSV-file as ASCII-value

Note:

- TC/LINK must be stopped and restarted before changes become effective.
- For common Registry entries, refer to the TC/LINK Manual.
- For link-specific Registry entries, refer to the corresponding link-specific manual