

Kofax Communication Server

TC/LINK Mapfile Technical Manual

Version: 10.3.0

Date: 2019-12-13

The KOFAX logo is rendered in a bold, blue, sans-serif font. The letters are thick and closely spaced, with a clean, modern aesthetic. The 'K' and 'O' 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	6
Chapter 2: Why MAPfiles?	7
The FAX/TELEX-Server.....	7
Mail Integration.....	7
KCS as a Message Switch.....	8
TC/LINK and the MAPFile.....	8
Chapter 3: Sections	9
Chapter 4: Objects Available for Use in the MAPfile	10
Mail Entry Fields.....	12
Intermediate Storage Objects (Variables).....	12
Registry Objects.....	13
Chapter 5: Rules	14
REM: Remark (Ignored).....	14
CHECKLINKQUEUE: Check If Address Is from the Foreign Mail System.....	14
COMPAREINT: Compare Integers.....	15
COMPARESTRING: Compare Strings.....	15
NCOMPARESTRING: Compare First n Characters in the Strings.....	16
COPY: Copy a Complete Address.....	16
COPYADDRESS_LIST: Take Address from a List.....	17
COPYADDRESS_STORE: Take Address from the Recipient Store.....	17
COPYAPPEND: Build a Destination String from 1 or More Source Parameters.....	18
COPYDIGITS: Copy Substring Starting with a Digit.....	19
COPYFMT: Copy a Formatted Source String to 1 or More Destination Fields.....	19
CREATE_OBJECT: Create a KCS User Profile or Address Book Entry.....	20
CREATE_DS_OBJECT: Create a Dirsync Enabled KCS User Profile or Address Book Entry.....	22
CVTSERVICE: Build KCS Address from Address Type Name and Number.....	23
FAIL_WITHOUT_NONDEL: Stop Processing of This Message and Disable Non-Delivery Notification.....	23
FROM_IMCEA: Translate IMCEA Encapsulated Address to TCOSS Address.....	24
ISOLATE_LOCALPART: Copy Local Part of SMTP Address.....	24
IS_COMPLETE_ADDRESS: Verify format of email address.....	25
ONLYTHIS: Ignore All Other Alternative Addresses.....	25
REMOVETHIS: Ignore Current Address.....	25
REPLACE: Replace a Specific Character by Another Character.....	25

SEARCH_ADDRTYPE: Build an Address from Service Name and Number String.....	26
SEARCH_LINKSCV: Build Prefix for Mail Originator Address.....	26
SEARCH_MISSINGSVC: Add Default Service.....	27
SEARCH_TCADDR_IN_DDA: Search KCS Service and Address in DDA Fields.....	27
SEARCH_TCSVC_IN_DDA: Search KCS Service in DDA Name.....	27
SEARCH_USER_ADDRESS: Replace User with a User Having a Certain Proxy Address.....	28
SEARCH_USER_ID: Replace User with a User from KCS Recipient Store.....	28
SET_ANY_DDA: Write Service and Number to Domain Defined Attribute.....	29
STRLWR: Convert to Lowercase.....	30
STRUPR: Convert to Uppercase.....	30
TO_IMCEA: Create IMCEA Encapsulated Address.....	30
UPDATE_X400FIELDS: Update X.400 Address Fields.....	31
UPDATE_X400TEXT: Update TS_X400_ADDR.....	31
SET_GUEST: Use Special Guest User.....	32
SEARCH_USERID_ADDRESS: Find KCS User ID by Proxy Address.....	32
VALID_SMTP: Check if all characters allowed for SMTP address.....	33
Chapter 6: What Do “ENTRY”s Do?.....	35
The ENTRY Mask.....	35
Chapter 7: Examples.....	37
The TC/LINK-FI Standard MAPfile, Fully Documented.....	37
Section TO_TC_ORIG.....	37
Section TO_TC_RECP.....	38
Section TO_MAIL_ORIG.....	39
Section TO_MAIL_RECP.....	39
The TC/LINK-MX Standard MAPfile, Fully Documented.....	40
Section TO_TC_ORIG.....	40
Section TO_TC_RECP.....	45
Section TO_MAIL_ORIG.....	52
Section TO_MAIL_RECP.....	54
The TC/LINK-LN Standard MAPfile, Fully Documented.....	55
Section TO_TC_ORIG.....	55
Section TO_TC_RECP.....	56
Section TO_MAIL_ORIG.....	60
Section TO_MAIL_RECP.....	60
The TC/LINK-SM Standard MAPfile, Fully Documented.....	61
Section TO_TC_ORIG.....	61
Section TO_TC_RECP.....	62
Section TO_MAIL_ORIG.....	64

Section TO_MAIL_RECIP.....	67
The TC/LINK-SJ Standard MAPfile, Fully Documented.....	67
Section TO_TC_ORIG.....	67
Section TO_TC_RECIP.....	68
Section TO_MAIL_ORIG.....	70
Section TO_MAIL_RECIP.....	71
The TC/LINK-X4 Standard MAPfile, Fully Documented.....	71
Section TO_TC_ORIG.....	71
Section TO_TC_RECIP.....	73
Section TO_MAIL_ORIG.....	76
Section TO_MAIL_RECIP.....	78
The TC/LINK-SC Standard MAPfile, Fully Documented.....	78
Section TO_TC_ORIG.....	78
Section TO_TC_RECIP.....	81
Section TO_MAIL_ORIG.....	86
Section TO_MAIL_RECIP.....	88
The TC/LINK-AC Standard MAPfile, Fully Documented.....	90
Section TO_TC_ORIG.....	90
Section TO_TC_RECIP.....	92
Section TO_MAIL_ORIG.....	93
Section TO_MAIL_RECIP.....	95
Customization Examples for TC/LINK-SM.....	98
Disable Address Book Lookup For Incoming Messages.....	98
TC/LINK-SM Should Check the Complete Email Address for Distribution.....	98
Routing to Different Fax Lines Based on Originator Address.....	99
Customization Examples for TC/LINK-FI.....	101
Personal Address Book Entries.....	101
Sending Copy Messages.....	102
Customization Example for TC/LINK-AC.....	103
Chapter 8: Restrictions / Additional Remarks.....	105

Chapter 1

Preface

There are five theoretical and one practical step to become a MAPfile guru:

- Why do I need MAPfiles?
- What sections are available?
- What objects are available?
- What RULES are available?
- Why do I need “ENTRY”s?
- Some examples to illustrate the whole thing!

Chapter 2

Why MAPfiles?

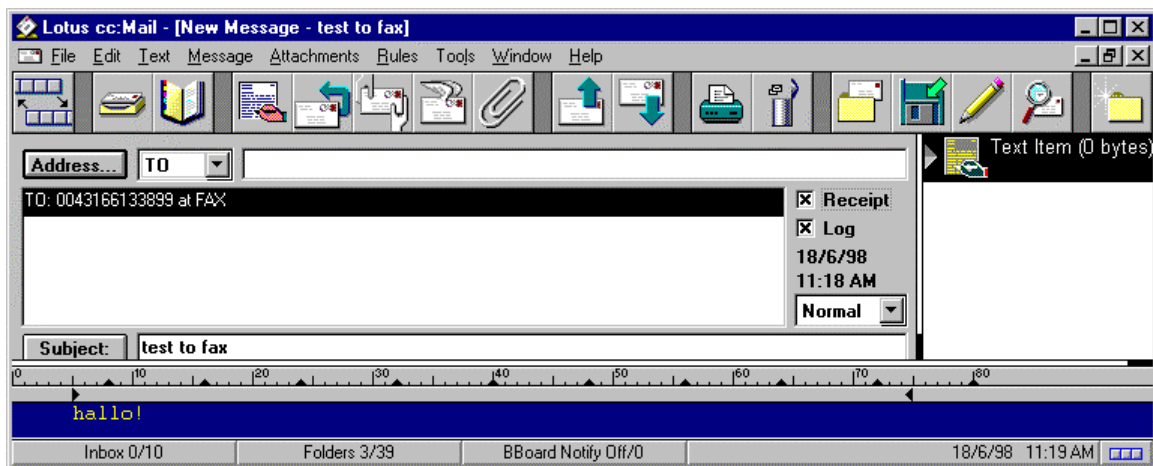
To understand why MAPfiles are necessary, we should have a look at the history of KCS product strategy.

The FAX/TELEX-Server

Originally, Kofax Communication Server was solely receiving/sending documents from/to FAX or TELEX. You simply had a KCS user on one hand and a FAX or TELEX number on the other hand. No need for MAPfiles ... (well, even if this is not 100% true: there were nn99 and rr99 that were involved in some address/number conversions / normalization / routing).

Mail Integration

Gradually, more and more proprietary email systems were connected to KCS. So, you were no longer sure to get only FAX/TELEX addresses on KCS – now, you had MS Mail, cc:Mail, Lotus Notes or even X.400 addresses on KCS! As long as all of them were used only to connect the respective mail system to FAX/TELEX, it was possible to use hardcoded address conversions (like in TC/GATE).



This example shows how a fax is addressed from cc:Mail. The destination service and the number are embedded into address fields provided by the mail system.

This works well for most services, but fails for more complex addresses, e.g. of X.400 type.

KCS as a Message Switch

As email to email communication got more and more importance, there was the request to use Kofax Communication Server as a message switch to be able to send from any connected mail system to any other. Now, for instance,

- How do I embed an MS Exchange address that way that it can be handled from the receiving Lotus Notes client?
- How can I make sure that replies will work under all circumstances?
- And how can I add some flexible way to customize addressing if a customer needs it?

That was the point where the MAPfile was invented as a part of the TC/LINK architecture. All TC/LINKs make use of its powerful features to translate any kind of addresses.

TC/LINK and the MAPFile

TC/LINK uses the MAPFile for address conversion.

The MAPfile defines how originator and recipient addresses shall be converted between KCS address format and the address format of the remote mail system.

TC/LINK must talk to two messaging systems: to the Kofax Communication Server and to the MTA of the remote mail systems:

- In a message posted to KCS, all addresses must be in KCS address format.
- In a message to the remote mail system, all addresses must be in a format that the mail MTA understands.
- Replies and notifications must find their way back to the message originator.
- Additionally, a mail user normally wants to use the send options defined in his KCS user profile, and wants to find his out-mail in his KCS Outbox. TC/LINK must find the KCS user profile for the originator of a mail message.
- The address format on the remote mail system can also consist of different components. The special link software is responsible for manipulating the mail message itself, this includes extracting an address from a message and inserting an address into a message.
- The addresses exchanged between the special link DLL and TC/LINK are mail system format addresses packed into a KCS address structure, in most cases into a SET_FREE_ADDRESS structure. Some links (e.g. TC/LINK-X4, TC/LINK-MX) also support an SET_X400_ADDRESS. In many cases, also the Service name holds addressing information.

This address only looks like a KCS address, but it cannot be routed by TCOSS, it is just used to pass information to/from TCLINK.

Chapter 3

Sections

TC/LINK uses the MAPfile for four types of conversions:

Addresses in messages from the mail system must be converted to KCS format:

- Originator and passive recipients must get an address that allows to send notifications and messages to them via TCLINK. If they have KCS shadow users, TC/LINK must replace the original mail address with the shadow user. (e.g. for default send options, rights etc.)
- Active recipients (like fax numbers etc.) must get an address which tells TCOSS where to send this message. As much information as possible must be passed from the mail system to KCS (e.g. information used in coversheets).

Addresses in messages from KCS must be converted to the mail system's address format:

- Originator and passive recipients must get an address that allows to send notifications and messages back to them via TCLINK.
- Active recipients must get an address that can be routed by the MTA of the mail system.

These conversion types are reflected in the MAPfile.

TC/LINK uses four MAPfile sections:

- "TO_TC_ORIG" (message to KCS: originator and passive recipients)
- "TO_TC_RECIP" (message to KCS: active recipients)
- "TO_MAIL_ORIG" (message to mail system: originator and passive recipients)
- "TO_MAIL_RECIP" (message to mail system: active recipients)

Chapter 4

Objects Available for Use in the MAPfile

There are many originator/recipient properties available for building an address. The following section will show you how to access the available fields in the MAPfile.

Most important fields.

Keyword	Explanation	Remarks
TS_RECIP_ID	KCS user short name	
TS_COMPANY	Company name	
TS_FULLNAME	Full name	
TS_XFIELD	Additional information	TC/LINK-SM: long display name
TS_DEPTM	Department	
TS_SALUTE	Salutation	
TS_FREETEXT	Comment	
TS_SERVICE	Service name	
TS_FAX_NUMBER	Fax number	Only for address type FAX
TS_FAX_ANSWERB	Fax answerback	Only for address type FAX
TS_TX_NUMBER	Telex number	Only for address type TELEX
TS_TX_ANSWERB	Telex answerback	Only for address type TELEX
TS_TTX_NUMBER	Teletex number	Only for address type TELETEX
TS_TTX_ANSWERB	Teletex answerback	Only for address type TELETEX
TS_TC_NODE	TCOSS node	Only for address type TC
TS_TC_USERID	KCS user ID	Only for address type TC
TS_FREE_ADDR	Free address string	Only for address type FREE

Additionally, the following fields can be used (usually not required).

Keyword	Explanation	Remarks
TS_CORREL_1 ... TS_CORREL_5	Correlation information strings	
TS_SECTION	The recipient store section	Usually "+TECH" (system address book)
INT_RECIP_ACTIVE	Active-flag (per-recipient)	
INT_DEL_TYPE	Delivery type	To, cc, bcc, auth

Keyword	Explanation	Remarks
INT_PRIORITY	Priority	High / normal / low
INT_TYPE	Recipient type	Recipient / user
INT_OWNERTYPE	Ownertype (for dirsync)	
INT_DIRSYNC_ALLOWED	Dirsync allowed	
INT_ADDR_ACTIVE	Active flag (per-address)	
TS_WHOLENUMBER	Complete number string, including channel	
TS_TRUENUMBER	Number string without channel	
TS_NUMBER	Number string without service-specific prefix	
TS_X400_ADDR	Complete X.400 address string	Only in X.400 addresstype
TS_X400_C	Country	Only in X.400 addresstype
TS_X400_AD	Administrative domain	Only in X.400 addresstype
TS_X400_PD	Private domain	Only in X.400 addresstype
TS_X400_ORG	Organisation	Only in X.400 addresstype
TS_X400_OU1	Organisational unit 1	Only in X.400 addresstype
TS_X400_OU2	Organisational unit 2	Only in X.400 addresstype
TS_X400_OU3	Organisational unit 3	Only in X.400 addresstype
TS_X400_OU4	Organisational unit 4	Only in X.400 addresstype
TS_X400_SNAME	Surname	Only in X.400 addresstype
TS_X400_GNAME	Givenname	Only in X.400 addresstype
TS_X400_INITLS	Initials	Only in X.400 addresstype
TS_X400_GENQUAL	Generation qualifier	Only in X.400 addresstype
TS_D1NAME ... TS_D4NAME	DDA names	Only in X.400 addresstype
TS_D1DATA ... TS_D4NAME	DDA values	Only in X.400 addresstype

Source and Destination fields

During address conversion, a source address (as it comes from the sending mail system) is converted into a target address (as it is accepted by the receiving mail system).

So, for sending to KCS (sections TO_TC_ORIG, TO_TC_RECIP), the source address is a mail system address packed e.g. into a TCOSS free address structure, and the destination address may be any address that TCOSS understands (even a complete KCS address book entry).

When sending to the Mail system (sections TO_MAIL_ORIG, TO_MAIL_RECIP), it is the other way round. The source address includes the typical KCS address components (+ address book attributes like Fullname, Company ...), whereas the destination address is a mail system address packed into a KCS address structure (e.g. a SET_FREE_ADDRESS containing an Internet address).

As both source and destination address are packed into KCS address structures, the objects listed above can be used for both of them.

To refer to elements of the source address (= unconverted address) use the object names with prefix "SRC." (e.g. SRC.TS_RECIP_ID).

For elements of the destination address (= converted address) use the object names with prefix "DST." (e.g. DST.TS_RECIP_ID).

Attention:

DST elements are initialized at the start of a new entry.

SRC elements are permanent for the whole address conversion. Do not change SRC elements !

Mail Entry Fields

Address mapping rules in sections TO_MAIL_ORIG and TO_MAIL_RECIP can access information from the mail entry (SET_ENTRY_MS_MAIL). All direct child objects of the mail entry with simple types (text, numeric, time) are available. Access is read-only, writing to these fields is not possible.

To refer to mail entry fields, use the prefix "PAR." followed by the name of the TCSI field, e.g. "PAR.TIME_ACTION", "PAR.TS_CORREL_4", "PAR.INT_NPAGE".

The TCSI field names are case-insensitive.

- Text fields can have up to 512 characters.
- Numeric fields are converted to decimal strings.
- Date and time fields are converted to strings in the format "YYMMDD:hhmmss", based on the time zone of the link user.

Please use the current version of the TCSI documentation (manual or html help) to find all possible child objects of the SET_ENTRY_MS_MAIL.

See [Sending Copy Messages](#) for an example.

Intermediate Storage Objects (Variables)

For intermediate storage of any data, there are two kinds of variables available.

Variables on section level:

They are addressed "ADVAR.n", where n ranges from 0 ... 9.

Example: ADVAR.0

Note

- Variables can hold a string of maximum 512 characters.
- The variables are initialized to an empty string at the start of a new map file section, i.e. when starting address mapping for an address!

Variables on entry level:

They are addressed “VAR.n”, where n ranges from 0 ... 9.

Example: VAR.0

Note

- Variables can hold a string of maximum 512 characters.
- The variables are initialized to an empty string at the start of any new entry!

Registry Objects

You can also access any objects located in the registry by simply typing “REG.<relative_name>”. <relative_name> means the name of the registry value relative to the registry key of the process.

Example for TC/LINK-SM: REG.TCLSM\SMLinkDomain will access the configured local domain name assigned to TC/LINK-SM.

Note Be careful when writing to the registry as this may impact your server configuration! If you attempt to write on a non-existing key, the key will be created. In difference to VAR objects, the registry entries are permanent (no initialized or cleared by the mapping process)

The MAPfile allows access to nearly all fields of the internal TCSI address structure, plus registry access, and a set of variables.

Chapter 5

Rules

In the section above, we learned a lot about the various objects available. Now we need to figure out how we can manipulate these objects.

Basically, it is always the same:

- You take one or more objects from the source side.
- You split, copy, convert and rearrange these objects by control of rules.
- Finally, you store the output to the destination side objects.

Note

- Every rule has 0 or more parameters.
- Each parameter starts in the first column of a new line.
- If a parameter contains spaces, it must be quoted (e.g. "parameter with space")

The following lists all available rules:

REM: Remark (Ignored)

No parameters.

Easy but important: Just like in MS-DOS batch files, all lines starting with REM are ignored by the mapping process. Use it to make your MAPfile more understandable.

CHECKLINKQUEUE: Check If Address Is from the Foreign Mail System

This rule has no parameters.

It succeeds if the source address uses one of the queues defined for this instance of TCLINK. Otherwise it fails. It was designed to find out passive mail recipients in messages from KCS to mail.

If you use the CHECKLINKQUEUE in an entry of section TO_MAIL_ORIG, the rest of the entry will only be done for passive recipients on the foreign mail system.

Example:

```
TO_MAIL_ORIG -----  
ENTRY  ----- (check for SMTP address; if available, take only this!)
```

```
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
CHECKLINKQUEUE
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_TRUENUMBER
ONLYTHIS
```

In the example, passive mail recipients get their real SMTP address, instead of an address consisting of fullname, service and number.

COMPAREINT: Compare Integers

Syntax

COMPAREINT	Rule command
Operator	EQU or NEQU
Integer_1	Comparison argument 1
Integer_2	Comparison argument 2

Example

```
COMPAREINT
EQU
SRC.INT_TYPE
0
```

Compares two integers (constant, or any object starting with INT, e.g. INT_PRIORITY). Depending on operator (EQU or NEQU), the rule succeeds if they are equal or different. If the rule fails, the current entry is skipped.

COMPARESTRING: Compare Strings

Syntax

COMPARESTRING	Rule command
Operator	EQU ... case-sensitive equal NEQU ... case-sensitive non-equal IEQU ... non-case-sensitive equal INEQU ... non-case-sensitive non-equal
String_1	Comparison argument 1
String_2	Comparison argument 2

Example

```
COMPARESTRING
EQU
```

```
SRC.TS_TC_USERID
"postmaster"
```

Compares two strings. Depending on operator (EQU/NEQU/IEQU/INEQU), the rule succeeds if they are equal or different. If the rule fails, the current entry is skipped.

NCOMPARESTRING: Compare First n Characters in the Strings

Syntax

NCOMPARESTRING	Rule command
Operator	EQU ... case-sensitive equal NEQU ... case-sensitive non-equal IEQU ... non-case-sensitive equal INEQU ... non-case-sensitive non-equal
String_1	Comparison argument 1
String_2	Comparison argument 2
Integer_N	Comparison argument 3

Example

```
NCOMPARESTRING
EQU
SRC.TS_TC_USERID
"pos"
3
```

Compares n characters of two strings. Depending on operator (EQU/NEQU/IEQU/INEQU), the rule succeeds if they are equal or different. If the rule fails, the current entry is skipped.

The example above succeeds for KCS user ids like "postmaster" or "poseidon" and fails for KCS user ids like "ponderosa" or "pastmaster"

COPY: Copy a Complete Address

Syntax

COPY	rule command
Source	source UN_PUBLIC_ADDRESS
Destination	destination UN_PUBLIC_ADDRESS

The COPY command is used if an address shall not be changed by address mapping.

Example

```
COPY
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
```


This example is the only usage that makes sense.

Note This rule does not copy the TS_SERVICE field.

COPYADDRESS_LIST: Take Address from a List

Syntax

COPYADDRESS_LIST	rule command
name	TCSI field or string with recipient name
adresstype	F for fax, T for telex, E for EMAIL (X400)
filename	full path name of file containing list

Example

```
COPYADDRESS_LIST
SRC.TS_TRUENUMBER
F
.\APPLI.LST
```

This rule searches in a text file for the address of a particular recipient. If the address is found, it is copied to the destination address. The syntax of the text file is:

```
Urecipientid
Ffaxaddress
Ttelexaddress
Ex400address
```

The first character of a line determines the type of information in this line. A line starting with 'U' contains the name of a recipient, a line starting with 'F' holds his fax address etc.

The second parameter of the COPYADDRESS_LIST command defines which address type shall be searched.

The rule is part of the TC/LINK-AC MAPFile but most probably not used.

Note This rule modifies DST.UN_PUBLIC_ADDRESS

COPYADDRESS_STORE: Take Address from the Recipient Store

Syntax

COPYADDRESS_STORE	rule command
Name	TCSI field or string with recipient name
Adresstype	F for fax, T for telex

This rule is used by TC/LINK-AC to find the fax (or telex) address of a shadow user. This address is needed for notifications to APPLI/COM.

Example

```
COPYADDRESS_STORE
SRC.TS_TRUENUMBER
F
```

The first parameter (“Name”) is the KCS user ID of the shadow user, the second parameter (“Addresstype”) tells which address is wanted (fax or telex).

The rule looks for the user’s recipient store and tries to find a matching address. If found, the address is written to DST.UN_PUBLIC_ADDRESS.

Note This rule modifies DST.UN_PUBLIC_ADDRESS

COPYAPPEND: Build a Destination String from 1 or More Source Parameters

Syntax

COPYAPPEND	Rule command
FORCE	Optional keyword
Destination	Destination TCSI field
Source1	First source component (TCSI field or string)
Source2	Second source component
...	And so on

TCLINK concatenates all source components to a string and writes this string to the destination field. If the rule contains the optional parameter FORCE, TC/LINK writes the destination string even if it is empty. Normally, COPYAPPEND does not write an empty destination string.

Example

```
COPYAPPEND
DST.TS_FREE_ADDR
"
SRC.TS_TC_USERID
"
"@
REG.General\ServerDomain
```

This command builds the Internet address (“mailbox@maildomain”) by copying the quoted KCS User ID, the “@” sign, and the configured serverdomain together.

Note Single quotation marks can be inserted by typing a single <“> on a separate line (like in the example)!

COPYDIGITS: Copy Substring Starting with a Digit

Syntax

COPYDIGITS	rule command
Service	TCSI variable or string holding target service name
Source	source TCSI variable or string
Destination	destination TCSI variable

This rule is used by TC/LINK-MX to isolate a fax or telex number from an MSMAIL address.

Example

```
COPYDIGITS
DST.TS_SERVICE
DST.TS_RECP_ID
DST.TS_RECP_ID
```

If the service parameter has address type FAX, TELEX or TELETEX, TC/LINK scans the source string for a substring starting with a digit. This substring is then copied to the destination field.

If no digit is found in the source string or if the address type is different, the rule copies the whole source string to the destination field.

COPYFMT: Copy a Formatted Source String to 1 or More Destination Fields

This rule can be used

- To separate a sequence of values into individual components. The values in the source string are separated by a specific character (or string). The components may be optional or mandatory.
- To check for occurrence of a specific character (or string) in a field.

Syntax

COPYFMT	Rule command
Source	TCSI field containing the formatted string
Formatstring	Specifies the components of the formatted string and the character(s) separating the components
Destination1	TCSI field receiving the first component
Destination2	TCSI field receiving the second component
...	and so on

The second parameter (formatstring) specifies a separator string and the optional and mandatory components. Mandatory components are represented by a '!' character, optional components are represented by a '?' character.

Formatstring examples:

“;?!” separator is “;” ; at least one occurrence required.

“ AT ?!” separator is “ AT “; occurrence optional.

Example

```
COPYFMT
SRC.TS_FREE_ADDR
"@!!"
SRC.TS_NUMBER
VAR.0
VAR.1
```

This command will succeed if there is at least one occurrence of “@” in the KCS address string of type “FREE”. The string left of the first “@” will be copied to VAR.0, the string right of it to VAR.1.

Note Characters ‘?’ and ‘!’ are not allowed in the separator string.

CREATE_OBJECT: Create a KCS User Profile or Address Book Entry

This rule can be used

- To create a KCS user profile or address book entry. The object is dirsync-disabled.
- Optionally, a per-user license is consumed for the new object.
- Optionally, an administrator receives a message about success or failure of the action.
- The new object can be used for further address mapping, e.g. as the sender of the message.

Syntax

CREATE_OBJECT	Rule command
SuccessMessageTemplate	Template for success message to admin, specified as Folder/Filename (e.g. TCLINK/MFPOK), or empty
ErrorMessageTemplate	Template for error message to admin, specified as Folder/Filename, or empty
RegisteredObject	Name for registration (e.g. MFP IP address), or empty if no registration needed
LicenseType	Numeric license type (e.g. 521 for TC/LINK-MFP), or empty if no registration needed.
Name	User name (referenced as \$Name\$ in template user)
TemplateName	Name of template user (e.g. MFPTEMPL)
DirsyncParameter1	Optional, value of dirsinc parameter 1 (referenced as \$1\$ in template user)
DirsyncParameter2	Optional, \$2\$
...	and so on ...

The algorithm first checks if the user already exists. The actions described below (license check, user profile writing, message to operator) are only done if the user does not yet exist on Kofax Communication Server.

The resulting user profile is not dirsync-enabled. It belongs to the same mail system as the user template.

The command succeeds if the KCS user already exists or has been created successfully (including the optional license registration).

Example

```
CREATE_OBJECT
"TCLINK/MFPOK"
"TCLINK/MFPERR"
SRC.TS_CORREL_5
521
SRC.TS_CORREL_5
MFPTEMPL
SRC.TS_CORREL_4
SRC.TS_CORREL_5
```

Dirsync parameters:

In the template user profile, variable \$Name\$ is a placeholder for the user name (Name parameter) and variables \$1\$, \$2\$ etc. are used as placeholders for the dirsync parameters 1, 2 etc.

Licensing:

If parameter RegisteredObject is not empty, a license check is done before creating the new user. This means that a registration is made with this name, for the specified license type.

If the registration fails, the user is not created.

Messages to the operator:

If user creation succeeds and the parameter SuccessMessageTemplate is not empty, the specified message template is used for creating a message.

If user creation fails and the parameter ErrorMessageTemplate is not empty, the specified message template is used for creating a message.

The following description refers to both message types (success and error):

The complete template message (including sender, recipients, sending options, content) is used as a basis for the new message.

In the message subject and text blocks, the following variables are substituted:

- \$Name\$ – name of the created user
- \$TimeUtc\$ – current date and time in GMT time (YYYY-MM-DD hh:mm:ss)
- \$TimeLocal\$ – current date and time in local time (YYYY-MM-DD hh:mm:ss)
- \$LicensesLeft\$ – number of remaining licenses
- \$Error\$ – internal error number (e.g. 621 for licensing error)
- \$LicenseOwner\$ – name used for registration (RegisteredObject)

Example for success message template:

Note:

A new MFP profile \$Name\$ has been created at \$TimeLocal\$.
 There are \$LicensesLeft\$ unused licenses left.
 Please configure a fax extension for the MFP device.
 The device URL is [http://\\$LicenseOwner\\$/](http://$LicenseOwner$/)

Text of the message sent to the administrator:**Note:**

A new MFP profile mfp1.company.com has been created at 2006-11-21 10:29:30.
 There are 6 unused licenses left.
 Please configure a fax extension for the MFP device.
 The device URL is [http://mfp1.company.com /](http://mfp1.company.com/)

CREATE_DS_OBJECT: Create a Dirsync Enabled KCS User Profile or Address Book Entry

This rule can be used

- To create a KCS user profile or address book entry. The object is dirsync-enabled.
- Optionally, a per-user license is consumed for the new object.
- Optionally, an administrator receives a message about success or failure of the action.
- The new object can be used for further address mapping, e.g. as the sender of the message.

Syntax

CREATE_OBJECT	Rule command
SuccessMessageTemplate	Template for success message to admin, specified as Folder/File name (e.g. TCLINK/MFP), or empty
ErrorMessageTemplate	Template for error message to admin, specified as Folder/File name, or empty
RegisteredObject	Name for registration (e.g. MFP IP address), or empty if no registration needed
LicenseType	Numeric license type (e.g. 521 for TC/LINK-MFP), or empty if no registration needed.
Name	User name (referenced as \$Name\$ in template user)
TemplateName	Name of template user (e.g. MFPTEMPL)
DirsyncParameter1	Optional, value of dirsync parameter 1 (referenced as \$1\$ in template user)
DirsyncParameter2	Optional, \$2\$
...	and so on ...

CREATE_DS_OBJECT creates an object that is dirsync-enabled. All other features are similar to CREATE_OBJECT.

CVTSERVICE: Build KCS Address from Address Type Name and Number

Syntax

CVTSERVICE	rule command
Service	TCSI var. or string with address type name
Part1	address part 1
Part2	address part 2

This rule is used by TC/LINK-SM to build an address from an address type name and two parameters part1 and part2.

Example

```
CVTSERVICE
DST.TS_SERVICE
DST.TS_CORREL_1
DST.TS_CORREL_2
```

Depending on the address type, an address is created and parts 1 and 2 are interpreted as follows:

Address type name	part 1	part 2
TOPCALL	user id	node id
FAX	fax number	answerback
TELEX	telex number	answerback
TTX	teletex number	answerback
FREE	free address	

Note

- The TCSI variables holding address type name, part 1 and part 2 are deleted !
- This rule modifies DST.UN_PUBLIC_ADDRESS

FAIL_WITHOUT_NONDEL: Stop Processing of This Message and Disable Non-Delivery Notification

Syntax

FAIL_WITHOUT_NONDEL	rule command

This rule is used by TC/LINK-MFP to cause an error if the MFP profile cannot be created. It has no parameters.

Example

```
FAIL_WITHOUT_NONDEL
```

FROM_IMCEA: Translate IMCEA Encapsulated Address to TCOSS Address

Syntax

FROM_IMCEA	rule command
Address	IMCEA encapsulated address

This rule is used by TC/LINK-MX7 to translate an IMCEA encapsulated address into a TCOSS address. It modifies DST.UN_PUBLIC_ADDRESS and also the recipient fields (company, department etc.)

The only parameter is the variable holding the IMCEA address.

The function uses routines in TNC_TNEF.DLL and relies on the registry values that exist only for TC/LINK-MX7.

Example

```
FROM_IMCEA
SRC.TS_FREE_ADDR
```

ISOLATE_LOCALPART: Copy Local Part of SMTP Address

Syntax

ISOLATE_LOCALPART	rule command
Source	TCSI variable or string with full SMTP address
Localpart	TCSI variable that receives local part of address
Domain	Optional. TCSI variable that receives domain part of address

This rule can be used to separate local part and domain of an SMTP address. If the SMTP address starts with “(”, the local part is the string within parentheses. Otherwise the local part is everything before the first @ character. The domain part is everything after the @ character.

Example

```
ISOLATE_LOCALPART
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
```


IS_COMPLETE_ADDRESS: Verify format of email address

Syntax

IS_COMPLETE_ADDRESS	rule command
Source	TCSI variable or string with full SMTP address

This rule can be used to verify whether the SMTP address is in valid email format or not.

Example

```
IS_COMPLETE_ADDRESS
SRC.TS_NUMBER
```

ONLYTHIS: Ignore All Other Alternative Addresses

This rule has no parameters.

All other alternative addresses of the current originator/recipient are ignored. Address mapping for this recipient (or originator) is finished.

Use this command to improve performance (the mapping process is stopped as soon as a valid address is found).

REMOVETHIS: Ignore Current Address

This rule has no parameters.

The current address is ignored; nothing is copied to the destination object.

You may use this command to improve performance (fast rejection of any invalid alternative addresses).

REPLACE: Replace a Specific Character by Another Character

Syntax:

REPLACE	Rule command
Source_and_destination	TCSI field
Char1	Character before replacement
Char2	Character after replacement

Example

```
REPLACE
SRC.TS_TC_USERID
```

```
"/"
";"
```

This command replaces all slashes in the KCS User ID by semicolons.

SEARCH_ADDRTYPE: Build an Address from Service Name and Number String

Syntax

SEARCH_ADDRTYPE	rule command
Service	TCSI field or string with service name
Number	TCSI field or string with number
SEP=separator Or SEPALL=separator	1 character, separates number and answerback (optional)

This rule builds a destination address according to service name and number string. It fails if the service does not exist on KCS or if the number format is invalid (in X400 addresses). The optional third parameter (separator) is a character that separates number and answerback.

If the third parameter is specified in syntax "SEPALL=separator", a separator at the end of the number string is removed.

Example

```
SEARCH_ADDRTYPE
SRC.TS_SERVICE
SRC.TS_FREE_ADDR
SEP=-
```

Note This rule modifies DST.TS_SERVICE and DST.UN_PUBLIC_ADDRESS

SEARCH_LINKSCV: Build Prefix for Mail Originator Address

This rule has no parameters. It is typically used in section TO_TC_ORIG, as the last rule of an entry converting mail originators without KCS shadow user.

Using SEARCH_LINKSVC guarantees correct routing of notifications.

The algorithm finds out the default link queue and the default image format (from TCLINK configuration). From these settings, it builds a queue name. The resulting queue name is used as a prefix of the resulting address. If there is a KCS service using this prefix, the resulting address will use this service (otherwise it will use service FREE).

Note This rule modifies DST.TS_SERVICE and DST.UN_PUBLIC_ADDRESS

SEARCH_MISSINGSVC: Add Default Service

This rule has no parameters. Some links use it in section TO_TC_RECIP, to find a default KCS service for sending.

If the destination address has no TS_SERVICE parameter, this rule causes TCLINK to write a default service to DST.TS_SERVICE. The name of the default service depends on the address type. Default services for all address types are configured at link setup.

Example

```
COPYFMT
SRC.TS_FREE_ADDR
"@!!"
DST.TS_FULLNAME
DST.TS_FAX_NUMBER
SEARCH_MISSINGSVC
```

Note This rule modifies DST.TS_SERVICE

SEARCH_TCADDR_IN_DDA: Search KCS Service and Address in DDA Fields

This rule has no parameters. It is part of the TC/LINK-AC MAPFile, but currently not used.

This rule expects that the source address is an X.400 address and shall be mapped to a KCS address. The real KCS address is hidden in one of the domain defined attributes. DDA x name is expected to contain the service, DDA x value is expected to contain the number.

The algorithm looks for a domain defined attribute of the source address, containing a KCS service in its name component. If found, a new destination address is built from the service and number.

Note This rule modifies DST.TS_SERVICE and DST.UN_PUBLIC_ADDRESS

SEARCH_TCSVC_IN_DDA: Search KCS Service in DDA Name

This rule has no parameters. It is part of the TC/LINK-AC MAPFile, but currently not used.

This rule expects that the source address is an X.400 address and shall be mapped to a KCS X.400 address. As every KCS address needs a service, the KCS service name is expected to be in one of the four DDA name fields. The algorithm looks for a valid KCS service name in one of the DDA name fields. If found, the destination address gets this KCS service + the X.400 address from the source address.

Note This rule modifies DST.TS_SERVICE and DST.UN_PUBLIC_ADDRESS

SEARCH_USER_ADDRESS: Replace User with a User Having a Certain Proxy Address

Syntax

SEARCH_USER_ADDRESS	rule command
Address	Mandatory, mail address of originator (without prefix)
Service	Mandatory, service for mail address (or "" if any service allowed)
Numchars	Optional: number of significant trailing characters in address

This rule is used to find a shadow user by one of his proxy addresses. A proxy address is an (active or inactive) address that holds the mail-system address for this user. This rule relies on new TCOSS functionality and fails with TCOSS versions below 7.22.00.

Example

```
TO_TC_ORIG -----
ENTRY ----- enhanced orig. mapping
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
SEARCH_USER_ADDRESS
SRC.TS_FREE_ADDR
""
```

If the Numchars parameter is specified, TC/LINK removes all but the last "Numchars" characters from the address. This feature is only used by TC/LINK-SC, for separating the DID from a fax number.

TC/LINK uses TCOSS functionality to look for a KCS user with the specified address and service. If the service is empty (" " in mapfile), any service is accepted.

If this rule succeeds, the whole target person is replaced by the entry from the KCS recipient store.

Note After SEARCH_USER_ADDRESS, no other address mapping rule is allowed in the same mapfile entry.

SEARCH_USER_ID: Replace User with a User from KCS Recipient Store

Syntax

SEARCH_USER_ID	rule command
ALL	Optional parameter ALL: user or recipient

This rule can be used to

- find a shadow user for a mail originator (by name)
- find an address in the KCS address book (with parameter ALL)

Example

```
ENTRY ----- (email orig. with TC equiv.given by UserId = "gname sname")
*
SET_X400_ADDRESS
*
SET_TC_ADDRESS
COPYAPPEND
DST.TS_RECIP_ID
SRC.TS_X400_GNAME
" "
SRC.TS_X400_SNAME
SEARCH_USER_ID
```

It expects the KCS user ID in field DST.TS_RECIP_ID. If DST.TS_SECTION also exists, it is used as an additional search criterion. With these criteria, TCLINK.EXE searches in the KCS recipient store.

With the optional parameter ALL, TCLINK searches for a KCS user or address book entry.

Without parameter ALL, TCLINK only searches for a KCS user.

If a matching recipient store entry was found, the whole target person is replaced by the entry from the KCS recipient store.

Note After SEARCH_USER_ID, no other rule is allowed in the same mapfile entry.

SET_ANY_DDA: Write Service and Number to Domain Defined Attribute

Syntax

SET_ANY_DDA	rule command
Service	TCSI field or string with service name
Number	Optional: TCSI field or string with number

This rule is part of the TC/LINK-AC MAPFile, but currently not used.

It expects that source and destination address are both X.400. It copies the source address to the destination address, and inserts a domain defined attribute containing service (as name) and number (as value). The number parameter is optional.

Note Modifies DST_UN_PUBLIC_ADDRESS

STRLWR: Convert to Lowercase

Syntax

STRLWR	Rule command
Source_and_destination	TCSI field

Example

```
STRLWR
SRC.TS_TC_USERID
```

Converts the content of a TCSI string field to lowercase (in the example: the KCS user ID).

STRUPR: Convert to Uppercase

Syntax

STRUPR	Rule command
Source_and_destination	TCSI field

Example

```
STRUPR
SRC.TS_TC_USERID
```

Converts the content of a TCSI string field to uppercase (in the example: the KCS user ID).

TO_IMCEA: Create IMCEA Encapsulated Address

Syntax

TO_IMCEA	rule command
Address	Destination field for IMCEA encapsulated address

This rule is used by TC/LINK-MX7 to translate a TCOSS address into an IMCEA encapsulated address. It reads SRC.UN_PUBLIC_ADDRESS and writes into the variable or field passed as parameter.

The only parameter is the variable or TCSI field that receives the IMCEA address.

The function uses routines in TNC_TNEF.DLL and relies on the registry values that exist only for TC/LINK-MX7.

Example

```
TO_IMCEA
DST.TS_FREE_ADDR
```

UPDATE_X400FIELDS: Update X.400 Address Fields

Syntax

UPDATE_X400FIELDS	Rule command
Style	Optional: N for Notes, T for TCOSS (default)

Example

```
ENTRY ----- (recipient is X400 in free addr)
*
SET_FREE_ADDRESS
EMAIL
SET_X400_ADDRESS
COPYAPPEND
DST.TS_X400_ADDR
SRC.TS_TRUENUMBER
UPDATE_X400FIELDS
```

This rule updates the fields of an X.400 address according to the string representation of the X.400 address (TS_X400_ADDR).

The optional parameter “Style” specifies the syntax of the X400 address string. Currently, only N (Lotus Notes style) and T (TCOSS style) are supported. If the UPDATE_X400FIELDS rule is used without parameter, TCLINK assumes TCOSS syntax.

Note The algorithm needs a destination address of type X.400. It evaluates field DST.TS_X400_ADDR and sets the individual fields of the X.400 address accordingly.

UPDATE_X400TEXT: Update TS_X400_ADDR

Syntax

UPDATE_X400TEXT	Rule command
Style	Parameter 1: N for Notes, T for TCOSS (default)

This rule updates the string representation (TS_X400_ADDR) of an X.400 address according to the individual X.400 fields. When sending via KCS to an X.400 system, TCOSS needs the address in field TS_X400_ADDR.

Example

```
COPY
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT
```

The optional parameter “Style” specifies the syntax of the resulting TS_X400_ADDR field. Currently, only N (Lotus Notes style) and T (TCOSS style) are supported. Without parameter “Style”, TCLINK uses TCOSS syntax.

Lotus Notes style:

/S=x/G=x/Q=x/I=x/C=x/A=x/P=x/O=x/OU1=x/OU2=x/OU3=x/OU4=x/DDA:x=y/DDA:x=y

TCOSS style:

C=x;A=x;P=x;O=x;OU=x;OU=x;OU=x;OU=x;S=x;GI=x;I=x;GE=x;D=(x;y)D=(x;y)

Note

- The algorithm needs a destination address is of type X.400. It modifies the DST.TS_X400_ADDR field, using the information from the other X.400 fields.
- The sequence of parameters is different for TCOSS and Lotus Notes style.

SET_GUEST: Use Special Guest User

Syntax

SET_GUEST	Rule command
Guest user id	mandatory

This rule can be used to set the guest user for a specific message, based on some components in the address.

Example: See SEARCH_USERID_ADDRESS

SEARCH_USERID_ADDRESS: Find KCS User ID by Proxy Address

Syntax

SEARCH_USERID_ADDRESS	Rule command
Destination for userid	Mandatory
Address	Mandatory, address without prefix
Service	Mandatory, service for mail address (or "" if any service allowed)
NumChars	Optional: number of significant trailing characters in address

This rule is used to find the ID of a shadow user by one of his proxy addresses. A proxy address is an (active or inactive) address that holds the mail-system address for this user. This rule relies on TCOSS functionality and fails with TCOSS versions below 7.22.00.

Whereas the SEARCH_USER_ADDRESS rule replaces the complete destination object, this rule only returns the ID of the user that was found.

The retrieved user ID can then be used for other purposes. In the example below, a search is done for a user with an address that matches the domain part of an email address. The user ID is then used as the guest user for this message.

The rule fails if no such user was found.

Example

```
ENTRY ----- enhanced orig. mapping for domain
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
ISOLATE_LOCALPART
SRC.TS_FREE_ADDR
VAR.1
VAR.2
SEARCH_USERID_ADDRESS
VAR.1
VAR.2
""
SET_GUEST
VAR.1
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
SEARCH_LINKSVC
```

VALID_SMTP: Check if all characters allowed for SMTP address

Syntax

VALID_SMTP	Rule command
Destination for userid	Mandatory
Mode	Mandatory, can be SAP or EXCHANGE
Number	Mandatory, the address to be checked

This rule checks if a sequence of characters can be used as the local part of an email address. TC/LINK-MX7 and TC/LINK-SC7 use this rule to check if an address can be converted into an email address without problems. Both links use an alternative addressing syntax if the address to be checked contains characters that are not supported by the SMTP protocol.

The rule fails if it finds characters that are not allowed in the local part of an SMTP address.

Example:

```
TO_MAIL_ORIG -----
ENTRY ----- internet syntax
*
*
SMTP
SET_FREE_ADDRESS
COMPARESTRING
EQU
REG.Exchange\SMTPSyntaxToMail
1
```

```
VALID_SMTP
EXCHANGE
SRC.TS_NUMBER
COPYAPPEND
DST.TS_FREE_ADDR
"
SRC.TS_NUMBER
"
@
SRC.TS_SERVICE
.
REG.TCLSM\SMLinkDomain
```

Chapter 6

What Do “ENTRY”s Do?

There are different types of originators / recipients on the KCS side, and there may be different address types on the remote mail system. For example:

- Alternative addresses (e.g. recipient from KCS address book)
- Holding only a simple address (e.g. originator from Internet)
- Multiple address types (e.g. FAX, FREE, KCS internal)
- Active and passive addresses
- KCS user profiles (may hold one or more active or inactive addresses)
- X.400 - styled address

It is nearly impossible to cover all of them with a single set of rules. Therefore, you can add multiple so-called “entries” to your MAPfile.

The mapping process takes an address, and tries to convert it according to the first entry. If it succeeds, everything is fine; if it fails, it will try the next entry, and so on. So, you can chain multiple possibilities for mapping by putting multiple entries.

Let’s jump into the syntax of an entry:

- Each entry starts with the keyword “ENTRY”.
- Any text following the keyword “ENTRY” on the same line is treated as a comment (ignored by the mapping process).
- The four lines following the keyword “ENTRY” give a mask indicating whether the following entry is applicable to the current address or not.
- If the current address does not match the mask, then the entry is skipped for that address. The next entry is tried instead.
- If no entry leads to success, TC/LINK will refuse to send the current message.
- If the current address matches the mask, the rules defined for the entry are executed.

The ENTRY Mask

Syntax:

ENTRY	Entry Keyword
Source service	Entry applies only to specified source service name
Source address type	Entry applies only to specified source service type (one of “*”, SET_FAX_ADDRESS, SET_TX_ADDRESS, SET_TTX_ADDRESS, SET_TC_ADDRESS, SET_FREE_ADDRESS SET_X400_ADDRESS)

ENTRY	Entry Keyword
Destination service	Destination address gets this service (can be modified by a rule)
Destination address type	Destination address gets this address type (can be modified by a rule)
Optional parameter	Only used for TC/LINK-AC (originator address type)

A wildcard (“*”) in any of the fields means that the specific entry is not checked.

Example:

```
ENTRY ----- Build internet address from TC type address
*
SET_TC_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_TC_USERID
"@company.com"
```

If the message contains a TC-type (KCS internal) address, it will match this entry (source address type = SET_TC_ADDRESS; source service name is not checked).

The resulting Internet address will be in the syntax “<UserId>@company.com”!

ENTRYS allow for separate RULES for different address types and address properties. They add a lot of flexibility for building a customized addressing syntax.

Chapter 7

Examples

Note The examples in this section are old and do not reflect the current KCS version!

The TC/LINK-FI Standard MAPfile, Fully Documented

- The TC/LINK-FI MAPfile is very simple, because the File Interface uses the TCSI object model. Thus, only minimal address mapping is needed.
- MAPFile name: ADRFI.MAP

Section TO_TC_ORIG

- Source addresses may be of any address type.
- Shadow user search only if enabled.

```
REM *****
REM Address mapping file for TC/LINK-FI
REM
REM History:
REM 1.07.02 Common Mapfile for shadowuser/no shadowuser (selected by registry)
REM 1.09.00 Advanced originator mapping for inactive TCFI aliases
REM *****
REM
REM -----
TO_TC_ORIG    originator of message to KCS
REM -----
REM -----
REM originator with KCS shadow user or KCS recipient
REM -----
ENTRY
*
SET_TC_ADDRESS
*
SET_TC_ADDRESS
COMPARESTRING    check configuration key UseShadowUser
IEQU
REG.Options\UseShadowUser
"yes"
COPYAPPEND    copy userid to DST (for SEARCH_USER_ID)
DST.TS_RECP_ID
SRC.TS_RECP_ID
SEARCH_USER_ID    search user (or addressbook entry) with this ID
ALL
REM -----
REM originator with KCS shadow user (enhanced originator mapping)
REM -----
ENTRY
*
```

```

SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COMPARESTRING    check if configured for shadow users
IEQU
REG.Options\UseShadowUser
"yes"
SEARCH_USER_ADDRESS    search user by address
SRC.TS_FREE_ADDR
""
REM or REG.Setup\ServiceFI\Name
REM -----
REM X400 originator without KCS shadow user (or shadow user disabled)
REM -----
ENTRY
*
SET_X400_ADDRESS
REG.TOPCALL\X400Service
SET_X400_ADDRESS    use default X400 service
COPY    just copy X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT    write string representation of X400 address
REM -----
REM other originator without KCS shadow user (or shadow user disabled)
REM -----
ENTRY
*
*
*
*
COPY    just copy address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND    and service
DST.TS_SERVICE
SRC.TS_SERVICE

```

Section TO_TC_RECIP

- Source addresses can be any address type.

```

REM -----
TO_TC_RECIP    recipient of message to KCS
REM -----
REM -----
REM KCS user or addressbook entry
REM -----
ENTRY
*
SET_TC_ADDRESS
*
SET_TC_ADDRESS
COMPARESTRING    proceed only if shadow users enabled
IEQU
REG.Options\UseShadowUser
"yes"
COPYAPPEND    for lookup, recipient id must be in dst
DST.TS_RECIP_ID
SRC.TS_RECIP_ID
SEARCH_USER_ID    look for KCS user or addressbook entry
ALL
REM -----

```

```

REM X400 recipient address (for TC/LINK-X4)
REM -----
ENTRY
*
SET_X400_ADDRESS
REG.TOPCALL\X400Service use default X400 service
SET_X400_ADDRESS
COPY just copy X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT and update the address string
REM -----
REM all other recipients (no conversion)
REM -----
ENTRY
*
*
*
*
COPY copy address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND copy service
DST.TS_SERVICE
SRC.TS_SERVICE

```

Section TO_MAIL_ORIG

- No conversion needed.

```

REM -----
TO_MAIL_ORIG originator of message to FileInterface
REM -----
REM -----
REM all originators: just copy address
REM -----
ENTRY
*
*
*
*
COPY copy address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND copy service
DST.TS_SERVICE
SRC.TS_SERVICE

```

Section TO_MAIL_RECIP

- No conversion needed.

```

REM -----
TO_MAIL_RECIP recipient of message to FileInterface
REM -----
REM -----
REM all recipients: just copy address
REM -----
ENTRY
*
*
*

```

```

*
COPY    copy address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND    copy service
DST.TS_SERVICE
SRC.TS_SERVICE

```

The TC/LINK-MX Standard MAPfile, Fully Documented

The TC/LINK-MX MAPfile is rather long and complicated, because:

- Every KCS address type corresponds to an Exchange address type
- Different addressing modes are available on the Exchange side (via templates, SMTP addresses, MSMail addresses, Outlook contacts)
- Originator mapping must be done for Exchange users AND MSMail users
- The standard addressing syntax has changed with version 1.09.00 and the old syntax is still supported
- MAPFile Name: ADR.MAP

Section TO_TC_ORIG

- Source addresses are of type SET_FREE_ADDRESS or SET_X400_ADDRESS.
- (Originator addresses are always Free addresses, passive recipients may have X400 addresses)
- SRC.TS_SERVICE holds the Exchange address type.
- SRC.TS_FREE_ADDR holds the Exchange address string (if not X400 address).
- SRC.TS_FULLNAME holds the display name.

```

REM -----
REM Address mapping for TC/LINK-MX
REM Version 1.07.02: - if config. var. MSNetWork:
REM                 first active TC orig. address is converted to MS
REM                 - additional parameter in COPYDIGITS
REM Version 1.07.04: - search TC userid for MS originators
REM Version 1.09.00: - advanced originator mapping
REM                 - Enhancement # 3741 (changed order of parms)
REM                 - support HL prefix (Exchange\HL must be HL)
REM                 - support Outlook contacts (with addrtype FAX)
REM Version 1.10.00: - fix error #4234 (internet syntax with @ inside)
REM                 : - fix error #4199 (displayname of mail users)
REM -----
REM -----
TO_TC_ORIG    originator of message to KCS
REM -----
REM -----
REM originator with KCS shadow user, enhanced originator mapping
REM -----
ENTRY
*    any service name(=Exchange address type) matches
SET_FREE_ADDRESS only FREE address matches
*    resulting service is not specified
SET_FREE_ADDRESS resulting address type is FREE (is changed later !)
COPYAPPEND    copy address string to variable 1
VAR.1    destination: VAR.1
SRC.TS_FREE_ADDR source: mail address string
SEARCH_USER_ADDRESS search for TC user with mail address

```



```

VAR.1 variable containing mail address
"" any service matches
REM or REG.Setup\ServiceEXCH\Name
REM -----
REM originator with KCS shadow user, via UserID (new syntax)
REM -----
ENTRY
REG.Exchange\TC registry key holding address type for KCS internal addresses
SET_FREE_ADDRESS only FREE address matches
* resulting service is not specified
SET_TC_ADDRESS resulting address type is TC (KCS internal)
COPYFMT separate address components
SRC.TS_FREE_ADDR
"#!???" 2 to 4 components, separated by #
DST.TS_RECP_ID KCS user name
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_TC_NODE node (optional)
SEARCH_USER_ID look for KCS user with this user name and node
REM -----
REM originator with KCS shadow user, via UserID (old syntax)
REM -----
ENTRY
REG.Exchange\TC registry key holding address type for TC addresses
SET_FREE_ADDRESS
*
SET_TC_ADDRESS
COPYFMT separate address components
SRC.TS_FREE_ADDR
";??!?" 2 to 4 components, separated by ,
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_RECP_ID KCS user name
DST.TS_TC_NODE node (optional)
SEARCH_USER_ID look for KCS user with this user name and node
REM -----
REM the following 6 entries find shadow users for MSMAIL users
REM -----
REM find shadow user for MS Mail Originator, Mapping type 2
REM User name is Mailbox/Postoffice/Network
REM -----
ENTRY ----- mapping type 2
MS applies only to address type MS
SET_FREE_ADDRESS
*
SET FREE ADDRESS
COMPARESTRING proceed only if mapping type is 2
EQU
REG.Exchange\MSOrigMapping
2
COPYFMT separate network, postoffice and mailbox
SRC.TS_FREE_ADDR
/!!!
VAR.3
VAR.2
VAR.1
COPYAPPEND change order of these components
DST.TS_RECP_ID
VAR.1
/
VAR.2
/
VAR.3
SEARCH_USER_ID look for KCS user with name Mailbox/Postoffice/Network

```

```
REM -----
REM find shadow user for MS Originator, mapping type 4
REM user name is Mailbox/Postoffice
REM -----
ENTRY
MS applies only to address type MS
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COMPARESTRING proceed only if mapping type is 4
EQU
REG.Exchange\MSOrigMapping
4
COPYFMT separate network, postoffice and mailbox
SRC.TS_FREE_ADDR
/!!!
VAR.3
VAR.2
VAR.1
COPYAPPEND build user name from mailbox and postoffice
DST.TS_RECP_ID
VAR.1
/
VAR.2
SEARCH_USER_ID search for KCS user with name Mailbox/Postoffice
REM -----
REM find shadow user for MS Originator, mapping type 8
REM user name is Mailbox
REM -----
ENTRY
MS applies only to address type MS
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COMPARESTRING proceed only if mapping type is 8
EQU
REG.Exchange\MSOrigMapping
8
COPYFMT separate network, postoffice and mailbox
SRC.TS_FREE_ADDR
/!!!
VAR.3
VAR.2
VAR.1
COPYAPPEND user name = mailbox
DST.TS_RECP_ID
VAR.1
SEARCH_USER_ID search for KCS user with name = Mailbox
REM -----
REM find shadow user for MS Originator, mapping type 2
REM X400 address, Mailbox/Postoffice/Network
REM sometimes, the MS address is wrapped into an X400 address
REM -----
ENTRY
X400 applies only to X400 addresses
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT isolate MS address (is in a DDA value)
SRC.TS_FREE_ADDR
/DDA:MS=?!
VAR.4
VAR.5
COMPARESTRING proceed only if mapping type is 2
```

```
EQU
REG.Exchange\MSOrigMapping
2
COPYFMT  separate network, postoffice and mailbox
VAR.5
/!!!!!!?  In an X400 address, / is escaped with a second /
VAR.3
*
VAR.2
*
VAR.1
*
COPYAPPEND  build user name from mailbox, postoffice and network
DST.TS_RECIP_ID
VAR.1
/
VAR.2
/
VAR.3
SEARCH_USER_ID  search for KCS user with this name
REM -----
REM  find shadow user for MS Originator, mapping type 4
REM  X400 address, Mailbox/Postoffice
REM -----
ENTRY
X400  applies only to address type X400
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT  isolate MS address (is in a DDA field)
SRC.TS_FREE_ADDR
/DDA:MS=?!
VAR.4
VAR.5
COMPARESTRING  proceed only if mapping type is 4
EQU
REG.Exchange\MSOrigMapping
4
COPYFMT  separate network, postoffice and mailbox
VAR.5
/!!!!!!?  In an X400 address, / is escaped with a second /
VAR.3
*
VAR.2
*
VAR.1
*
COPYAPPEND  build KCS user name from postoffice and mailbox
DST.TS_RECIP_ID
VAR.1
/
VAR.2
SEARCH_USER_ID  search for KCS user with this name
REM -----
REM  find shadow user for MS Originator, mapping type 8
REM  X400 address, Mailbox
REM -----
ENTRY
X400  applies only to address type X400
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT  isolate MS address (is in a DDA value)
SRC.TS_FREE_ADDR
```

```

/DDA:MS=?!
VAR.4
VAR.5
COMPARESTRING proceed only if mapping type is 8
EQU
REG.Exchange\MSOrigMapping
8
COPYFMT separate network, postoffice and mailbox
VAR.5
/!!!!!!? In an X400 address, / is escaped by a second /
VAR.3
*
VAR.2
*
VAR.1
*
COPYAPPEND name = mailbox
DST.TS_RECIP_ID
VAR.1
SEARCH_USER_ID search for a KCS user with this name
REM -----
REM If no shadow user found, take originator address + link queue
REM Take originator's display name as fullname
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND build address from address type and address string
DST.TS_FREE_ADDR
SRC.TS_SERVICE
": "
SRC.TS_FREE_ADDR
COMPARESTRING proceed only if display name (TS_FULLNAME) not empty
NEQU
SRC.TS_FULLNAME
""
COPYAPPEND fullname = display name
DST.TS_FULLNAME
SRC.TS_FULLNAME
SEARCH_LINKSVC add link queue (or service for it) to address
REM -----
REM If no shadow user found, take originator address + link queue
REM Take originator's address as fullname
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND build address from address type and address string
DST.TS_FREE_ADDR
SRC.TS_SERVICE
": "
SRC.TS_FREE_ADDR
COMPARESTRING proceed only if display name is empty
EQU
SRC.TS_FULLNAME
""
COPYAPPEND fullname = complete address
DST.TS_FULLNAME
SRC.TS_FREE_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address

```

Section TO_TC_RECIP

- Source addresses are either SET_FREE_ADDRESS or SET_X400_ADDRESS.
- SRC.TS_SERVICE holds the Exchange address type.
- SRC.TS_FREE_ADDR holds the Exchange address string (if not X400)

```

REM -----
TO_TC_RECIP    recipient of message to KCS
REM -----
ENTRY        recipient is SMTP address (syntax number@service.domain)
SMTP
SET_FREE_ADDRESS
*
*
ISOLATE LOCALPART separate local part and domain part
SRC.TS_FREE_ADDR
VAR.1    local part
VAR.2    domain part
COPYFMT isolate service in domain part
VAR.2    domain part
"!?"
VAR.3    service
*
COPYAPPEND shortname = number
DST.TS_RECIP_ID
VAR.1
SEARCH_ADDRTYPE build address from service and number
VAR.3
VAR.1
SEP=.    number may contain answerback (separated by .)
REM -----
REM recipient has MS address (company, name, number specified)
REM -----
ENTRY        recipient is MS address (company, name and number)
MS
SET_FREE_ADDRESS
*
*
COPYFMT isolate service (from postoffice)
SRC.TS_FREE_ADDR
"/?!?"
*
DST.TS_SERVICE
*
COPYFMT isolate address (from alias, may be within ``)
SRC.TS_RECIP_ID
"'?!?"
*
DST.TS_RECIP_ID shortname = address
*
COPYFMT separate address components: company#fullname#number
DST.TS_RECIP_ID
"#?!?"
*
DST.TS_COMPANY
DST.TS_FULLNAME
DST.TS_RECIP_ID shortname = number
SEARCH_ADDRTYPE build address from service and number
DST.TS_SERVICE
DST.TS_RECIP_ID
SEP=,    number may contain answerback (separated by ,)

```

```
REM -----
REM recipient has MS address (company, name specified)
REM -----
ENTRY    recipient is MS address with name and number
MS
SET_FREE_ADDRESS
*
*
COPYFMT isolate service (from postoffice)
SRC.TS_FREE_ADDR
"/?!?"
*
DST.TS_SERVICE
*
COPYFMT isolate address (from alias, may be within '')
SRC.TS_RECIP_ID
"'?!?"
*
DST.TS_RECIP_ID
*
COPYFMT separate address components: name#number
DST.TS_RECIP_ID
"#!!"
DST.TS_FULLNAME
DST.TS_RECIP_ID shortname = number
SEARCH_ADDRTYPE build address from service and number
DST.TS_SERVICE
DST.TS_RECIP_ID
SEP=, number may contain answerback (separated by ,)
REM -----
REM recipient has MS address (number specified, starts with digits)
REM -----
ENTRY    recipient address type is MS
MS
SET_FREE_ADDRESS
*
*
COPYFMT isolate service (from postoffice)
SRC.TS_FREE_ADDR
"/?!?"
*
DST.TS_SERVICE
*
COPYFMT isolate address (from alias, may be within '')
SRC.TS_RECIP_ID
"'?!?"
*
DST.TS_RECIP_ID
*
COPYDIGITS isolate number from address
DST.TS_SERVICE (fax, tlx, ttx: number starts with digit)
DST.TS_RECIP_ID
DST.TS_RECIP_ID
SEARCH_ADDRTYPE build address with service and number
DST.TS_SERVICE
DST.TS_RECIP_ID
SEP=, number may contain answerback (separated by ,)
REM -----
REM recipient has SMTP address (for TC/LINK-SM, Mailbox@Maildomain)
REM -----
ENTRY
SMTP    only for address type SMTP
SET_FREE_ADDRESS
*
```

```

SET_FREE_ADDRESS
SEARCH_ADDRTYPE build address (use default SMTP service)
REG.TOPCALL\SMTPService
SRC.TS_FREE_ADDR
REM -----
REM recipient has X.400 address (for TC/LINK-X4)
REM -----
ENTRY
X400    only for address type X400
SET_X400_ADDRESS comes as X400 address
REG.TOPCALL\X400Service use default X400 service
SET_X400_ADDRESS create X400 address
COPY    copy X400 address (unchanged)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT build X400 address string
REM -----
REM recipient has Lotus Notes address (for TC/LINK-LN, new syntax)
REM -----
ENTRY
REG.Exchange\NOTES only for configured Notes address type
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND use default NOTES service
DST.TS_SERVICE
REG.TOPCALL\NOTESService (may be changed below)
COPYFMT isolate address components
SRC.TS_FREE_ADDR
"#!?????" (separated by #)
DST.TS_FREE_ADDR
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
SEARCH_ADDRTYPE build address
DST.TS_SERVICE
DST.TS_FREE_ADDR
REM -----
REM recipient has Lotus Notes address (old syntax with ; as separator)
REM -----
ENTRY
REG.Exchange\NOTES only for configured Notes address type
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND use default NOTES service
DST.TS_SERVICE
REG.TOPCALL\NOTESService (may be changed below)
COPYFMT isolate address components
SRC.TS_FREE_ADDR
";??!????" (separated by ;)
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_FREE_ADDR
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
SEARCH_ADDRTYPE
DST.TS_SERVICE
DST.TS_FREE_ADDR

```

```

REM -----
REM FAX recipient created from Outlook contact)
REM -----
ENTRY
REG.Exchange\FAX  only for configured Fax address type
SET_FREE_ADDRESS
*
SET_FAX_ADDRESS becomes Fax address
COPYFMT  isolate address components
SRC.TS_FREE_ADDR
"@!!"  (displayname@number)
DST.TS_FULLNAME
DST.TS_FAX_NUMBER
COPYAPPEND  shortname = fax number
DST.TS_REC_P_ID
DST.TS_FAX_NUMBER
SEARCH_MISSINGSVC take default fax service
REM -----
REM FAX recipient created via new address template, for special customer
REM (only fax numbers with prefix HL are routed via TC/LINK-MX,
REM must remove the HL prefix)
REM -----
ENTRY
REG.Exchange\FAX  only for configured Fax address type
SET_FREE_ADDRESS
*
SET_FAX_ADDRESS address type becomes Fax
COMPARESTRING  only if feature is configured
EQU
REG.Exchange\HL
"HL"
COPYFMT  separate address components
SRC.TS_FREE_ADDR
"#!???"  new syntax: separator is '#'
DST.TS_FAX_NUMBER
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_FAX_ANSWERB
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
COPYFMT  remove HL prefix from number
DST.TS_FAX_NUMBER
"L!!"
*
DST.TS_FAX_NUMBER
COPYAPPEND  shortname = fax number
DST.TS_REC_P_ID
DST.TS_FAX_NUMBER
SEARCH_MISSINGSVC add link queue (or service) to number
REM -----
REM FAX recipient (created by template, new syntax)
REM -----
ENTRY
REG.Exchange\FAX  only for configured fax address type
SET_FREE_ADDRESS
*
SET_FAX_ADDRESS
COPYFMT  isolate address components
SRC.TS_FREE_ADDR
"#!???"  (separated by #)
DST.TS_FAX_NUMBER
DST.TS_FULLNAME

```



```
DST.TS_SERVICE
DST.TS_FAX_ANSWERB
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
COPYAPPEND shortname = fax number
DST.TS_REC_P_ID
DST.TS_FAX_NUMBER
SEARCH_MISSINGSVC if no service specified: take default
REM -----
REM FAX recipient (created by template, old syntax)
REM -----
ENTRY
REG.Exchange\FAX only for configured Fax address type
SET_FREE_ADDRESS
*
SET_FAX_ADDRESS
COPYFMT isolate address components
SRC.TS_FREE_ADDR
";?!!?????" (separated by ;)
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_FAX_NUMBER
DST.TS_FAX_ANSWERB
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
COPYAPPEND shortname = fax number
DST.TS_REC_P_ID
DST.TS_FAX_NUMBER
SEARCH_MISSINGSVC if no service specified: take default
REM -----
REM KCS recipient (created by template, new syntax)
REM -----
ENTRY
REG.Exchange\TC only for configured TC address type
SET_FREE_ADDRESS
*
SET_TC_ADDRESS
COPYFMT isolate address components
SRC.TS_FREE_ADDR
"#!???????" (separated by #)
DST.TS_TC_USERID
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_TC_NODE
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
COPYAPPEND shortname = user id
DST.TS_REC_P_ID
DST.TS_TC_USERID
SEARCH_MISSINGSVC if service not specified: take default
REM -----
REM KCS recipient (created by template, old syntax)
REM -----
ENTRY
REG.Exchange\TC only for configured TC address type
SET_FREE_ADDRESS
*
SET_TC_ADDRESS
```

```

COPYFMT isolate address components
SRC.TS FREE ADDR
";?!?!!?????" (separated by ;)
DST.TS FULLNAME
DST.TS SERVICE
DST.TS TC USERID
DST.TS TC NODE
DST.TS COMPANY
DST.TS DEPTM
DST.TS SALUTE
DST.TS FREETEXT
COPYAPPEND shortname = user id
DST.TS RECP_ID
DST.TS TC USERID
SEARCH_MISSINGSVC if service not specified: take default
REM -----
REM TELEX recipient (created by template, new syntax)
REM -----
ENTRY
REG.Exchange\TX only for configured Telex address type
SET_FREE_ADDRESS
*
SET_TX_ADDRESS
COPYFMT isolate address components
SRC.TS FREE_ADDR
"#!?!?!!?????" (separated by #)
DST.TS TX_NUMBER
DST.TS FULLNAME
DST.TS SERVICE
DST.TS TX_ANSWERB
DST.TS COMPANY
DST.TS DEPTM
DST.TS SALUTE
DST.TS FREETEXT
COPYAPPEND shortname = telex number
DST.TS RECP_ID
DST.TS TX_NUMBER
SEARCH_MISSINGSVC if service not specified: take default Telex service
REM -----
REM Telex recipient (created by template, old syntax)
REM -----
ENTRY
REG.Exchange\TX only for configured Telex address type
SET_FREE_ADDRESS
*
SET_TX_ADDRESS
COPYFMT isolate address components
SRC.TS FREE_ADDR
";?!?!!?????" (separated by ;)
DST.TS FULLNAME
DST.TS SERVICE
DST.TS TX_NUMBER
DST.TS TX_ANSWERB
DST.TS COMPANY
DST.TS DEPTM
DST.TS SALUTE
DST.TS FREETEXT
COPYAPPEND shortname = telex number
DST.TS RECP_ID
DST.TS TX_NUMBER
SEARCH_MISSINGSVC if service not specified: take default Telex service
REM -----
REM recipient with Free address (created by template, new syntax)
REM -----

```

```

ENTRY
REG.Exchange\FREE only for configured Free address type
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT isolate address components
SRC.TS_FREE_ADDR
"#!?????" (separated by #)
DST.TS_FREE_ADDR
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
COPYAPPEND shortname = free address
DST.TS_RECP_ID
DST.TS_FREE_ADDR
SEARCH_MISSINGSVC if no service specified: use default
REM -----
REM recipient with Free address (created by template, old syntax)
REM -----
ENTRY
REG.Exchange\FREE only for configured Free address type
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT isolate address components
SRC.TS_FREE_ADDR
";??!????" (separated by ;)
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_FREE_ADDR
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
COPYAPPEND shortname = free address
DST.TS_RECP_ID
DST.TS_FREE_ADDR
SEARCH_MISSINGSVC if no service specified: use default
REM -----
REM KCS addressbook entry (created by template, new syntax)
REM -----
ENTRY
REG.Exchange\RECIPIENT only for configured address type
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT isolate address components
SRC.TS_FREE_ADDR
"#!?" (name#addressbook)
DST.TS_RECP_ID
DST.TS_SECTION
SEARCH_USER_ID search addressbook entry
ALL (user or recipient)
REM -----
REM KCS addressbook entry (created by template, old syntax)
REM -----
ENTRY
REG.Exchange\RECIPIENT only for configured address type
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS

```

```

COPYFMT isolate address components
SRC.TS_FREE_ADDR
";!?" (name;addressbook)
DST.TS_RECIP_ID
DST.TS_SECTION
SEARCH_USER_ID search addressbook entry
ALL (user or recipient)

```

Section TO_MAIL_ORIG

- Destination address must be SET_FREE_ADDRESS or SET_X400_ADDRESS.
- DST.TS_SERVICE must be filled with Exchange address type.
- DST.TS_FREE_ADDR must be filled with Exchange address string (if not X400)

```

REM -----
TO_MAIL_ORIG   originator of message to Exchange
REM -----
REM -----
REM converted to MSMAIL syntax (active addresses)
REM (optional, if MSNetwork defined)
REM -----
ENTRY
*   for all types of addresses
*
MS   address type becomes MS
SET FREE_ADDRESS
COPYAPPEND read MSNetwork configuration key
VAR.1
REG.Exchange\MSNetwork
COMPARESTRING proceed only if MSNetwork not empty !
NEQU
VAR.1
""
COMPAREINT  proceed only if address is active !
EQU
SRC.INT_ADDR_ACTIVE
1
COPYAPPEND build MS address string
DST.TS_FREE_ADDR (MSNetwork/Service/dummy)
VAR.1
/
SRC.TS_SERVICE
/
Dummy
COPYAPPEND shortname = address (shortname becomes Alias)
DST.TS_RECIP_ID
SRC.TS_NUMBER
ONLYTHIS ignore all other addresses of this person
REM -----
REM converted to MSMAIL syntax (ignore inactive addresses)
REM (optional, if MSNetwork defined)
REM -----
ENTRY
*   for all types of addresses
*
MS   address type becomes MS
SET FREE_ADDRESS
COPYAPPEND read MSNetwork configuration key
VAR.1
REG.Exchange\MSNetwork
COMPARESTRING proceed only if MSNetwork is not empty
NEQU

```

```

VAR.1
""
COMPAREINT  proceed only if address is inactive
NEQU
SRC.INT_ADDR_ACTIVE
1
REMOVETHIS  remove inactive address
REM -----
REM SMTP originator (from default SMTP service)
REM -----
ENTRY
REG.TOPCALL\SMTPService only from default SMTP service
SET_FREE_ADDRESS
SMTP      address type becomes SMTP
SET_FREE_ADDRESS
COPYAPPEND  copy address (unchanged)
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
REM -----
REM NOTES originator (from default NOTES service)
REM -----
ENTRY
REG.TOPCALL\NOTESService only from default NOTES service
SET_FREE_ADDRESS
REG.Exchange\NOTES address type (as configured for NOTES)
SET_FREE_ADDRESS
COPYAPPEND  build address string
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR (address#fullname#service)
""
SRC.TS_FULLNAME
""
SRC.TS_SERVICE
REM -----
REM Fax originator
REM -----
ENTRY
*
SET_FAX_ADDRESS only for Fax addresses
REG.Exchange\FAX address type (as configured for FAX)
SET_FREE_ADDRESS
COPYAPPEND  build address
DST.TS_FREE_ADDR (number#fullname#service#answerback)
REG.Exchange\HL
SRC.TS_FAX_NUMBER
""
SRC.TS_FULLNAME
""
SRC.TS_SERVICE
""
SRC.TS_FAX_ANSWERB
REM -----
REM Telex originator
REM -----
ENTRY
*
SET_TX_ADDRESS only for Telex addresses
REG.Exchange\TX address type (as configured for Telex)
SET_FREE_ADDRESS
COPYAPPEND  build address
DST.TS_FREE_ADDR (number#fullname#service#answerback)
SRC.TS_TX_NUMBER
""
SRC.TS_FULLNAME

```

```

"#"
SRC.TS_SERVICE
"#"
SRC.TS_TX_ANSWERB
REM -----
REM KCS originator
REM -----
ENTRY
*
SET_TC_ADDRESS only for KCS internal addresses
REG.Exchange\TC address type (as configured for TC addresses)
SET FREE ADDRESS
COPYAPPEND build address
DST.TS_FREE_ADDR
SRC.TS_TC_USERID (userid#fullname#service#node)
"#"
SRC.TS_FULLNAME
"#"
SRC.TS_SERVICE
"#"
SRC.TS_TC_NODE
REM -----
REM originator with Free address
REM -----
ENTRY
*
SET_FREE_ADDRESS only for Free addresses
REG.Exchange\FREE address type (as configured for Free addresses)
SET FREE ADDRESS
COPYAPPEND build address
DST.TS_FREE_ADDR (address#fullname#service)
SRC.TS_FREE_ADDR
"#"
SRC.TS_FULLNAME
"#"
SRC.TS_SERVICE
REM -----
REM X400 originator
REM -----
ENTRY
*
SET_X400_ADDRESS only for X400 addresses
X400 address type becomes X400
SET_X400_ADDRESS
COPY copy X400 address (unchanged)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS

```

Section TO_MAIL_RECIP

- Destination address must be SET_FREE_ADDRESS.
- DST.TS_SERVICE must be filled with Exchange address type.
- DST.TS_FREE_ADDR must be filled with Exchange address string.

```

REM -----
TO_MAIL_RECIP recipient of message to Exchange
REM -----
REM -----
REM only 1 entry: service = address type, free address = address string
REM -----
ENTRY
*

```



```

SRC.TS_RECIP_ID
SEARCH_USER_ID  search user with this name
REM -----
REM originators without KCS shadow users
REM -----
ENTRY
NOTES
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND  copy NOTES address
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
SEARCH_LINKSVC  add link queue (or service for it) to address

```

Section TO_TC_RECIP

- Source addresses are always SET_FREE_ADDRESS.
- SRC.TS_SERVICE is either X400 or the service that shall be used for sending.
- If SRC.TS_SERVICE is X400, TCLINK uses the default X400 service
- SRC.TS_FREE_ADDR holds the address string, in various syntax types

```

REM -----
TO_TC_RECIP  recipient of message to KCS
REM -----
REM -----
REM recipient with X400 address (NOTES syntax)
REM -----
ENTRY
X400  only if service is X400
SET_FREE_ADDRESS
REG.TOPCALL\X400Service service becomes default X400 service
SET X400 ADDRESS
COPYAPPEND  copy address string
DST.TS_X400_ADDR
SRC.TS_FREE_ADDR
UPDATE_X400FIELDS fill X400 fields according to X400 string
N  (converting it from Notes to KCS X400 format)
REM -----
REM recipient with X400 address (modified KCS syntax)
REM (dots instead of semicolons)
REM -----
ENTRY
X400  only if service is x400
SET_FREE_ADDRESS
REG.TOPCALL\X400Service service becomes default X400 service
SET X400 ADDRESS
COPYAPPEND  copy address string
DST.TS_X400_ADDR
SRC.TS_FREE_ADDR
REPLACE  replace dots by semicolons
DST.TS_X400_ADDR
.
;
UPDATE_X400FIELDS fill X400 fields according to X400 string
REM -----
REM recipient in syntax "fullname AT company@number"
REM -----
ENTRY
*
SET_FREE_ADDRESS

```



```
*
*
COPYFMT check for " AT " and isolate fullname
SRC.TS_FREE_ADDR
" AT !!"
DST.TS_FULLNAME
VAR.1
COPYFMT separate company from number (in rest)
VAR.1
"@!!"
DST.TS_COMPANY
DST.TS_FREE_ADDR
COPYAPPEND copy service
DST.TS_SERVICE
SRC.TS_SERVICE
SEARCH_ADDRTYPE build address from service and number
DST.TS_SERVICE
DST.TS_FREE_ADDR
SEP=- (number may contain answerback, separated by -)
REM -----
REM SMTP recipients (one of 10 configurable SMTP services)
REM address syntax is "fullname#address"
REM -----
ENTRY
REG.TOPCALL\SMTPService only for service configured in TOPCALL\SMTPService
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService1 only for service configured in TOPCALL\SMTPService1
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService1
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService2 only for service configured in TOPCALL\SMTPService2
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService2
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService3 only for service configured in TOPCALL\SMTPService3
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService3
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
```

```
REG.TOPCALL\SMTPService4 only for service configured in TOPCALL\SMTPService4
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService4
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService5 only for service configured in TOPCALL\SMTPService5
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService5
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService6 only for service configured in TOPCALL\SMTPService6
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService6
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService7 only for service configured in TOPCALL\SMTPService7
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService7
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService8 only for service configured in TOPCALL\SMTPService8
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService8
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
ENTRY
REG.TOPCALL\SMTPService9 only for service configured in TOPCALL\SMTPService9
SET_FREE_ADDRESS
REG.TOPCALL\SMTPService9
SET_FREE_ADDRESS
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
REM -----
REM recipient in syntax "fullname@address"
REM -----
ENTRY
*
```

```
SET_FREE_ADDRESS
*
*
COPYFMT separate fullname from address
SRC.TS_FREE_ADDR
"@!!"
DST.TS_FULLNAME
DST.TS_FREE_ADDR
COPYAPPEND copy service
DST.TS_SERVICE
SRC.TS_SERVICE
SEARCH_ADDRTYPE build address from service and address
DST.TS_SERVICE
DST.TS_FREE_ADDR
SEP=- address may contain answerback (separated by -)
REM -----
REM recipient in syntax "fullname AT company#address"
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
*
COPYFMT check for " AT " and isolate fullname
SRC.TS_FREE_ADDR
" AT !!"
DST.TS_FULLNAME
VAR.1
COPYFMT check for "#" and separate company and address
VAR.1
"#!!"
DST.TS_COMPANY
DST.TS_FREE_ADDR
COPYAPPEND copy service
DST.TS_SERVICE
SRC.TS_SERVICE
SEARCH_ADDRTYPE build address from service and address
DST.TS_SERVICE
DST.TS_FREE_ADDR
SEP=- address may contain answerback (separated by -)
REM -----
REM recipient in syntax "fullname#address"
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
*
COPYAPPEND copy service
DST.TS_SERVICE
SRC.TS_SERVICE
COPYFMT separate fullname and address
SRC.TS_FREE_ADDR
#?!
DST.TS_FULLNAME
DST.TS_FREE_ADDR
SEARCH_ADDRTYPE build address from service and address
DST.TS_SERVICE
DST.TS_FREE_ADDR
SEP=- address may contain answerback (separated by -)
```

Section TO_MAIL_ORIG

- Destination address must always be a SET_FREE_ADDRESS.
- DST.TS_SERVICE must be either X400 or the originator service.
- DST.TS_FREE_ADDR must hold the address.

```

REM -----
TO_MAIL_ORIG   originator of message to NOTES
REM -----
REM -----
REM X400 originator
REM -----
ENTRY
*
SET X400_ADDRESS only for X400 addresses
X400   service becomes X400
SET X400_ADDRESS (will be changed later by rule)
COPY   copy X400 address (must be in DST for next rule)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT build X400 string in Lotus Notes syntax
N
SEARCH_ADDRTYPE build free address from X400 string
REG.TOPCALL\FreeService default free service (overwritten later)
DST.TS_X400_ADDR
COPYAPPEND destination service = X400
DST.TS_SERVICE
X400
REM -----
REM all other originators
REM -----
ENTRY
*
*
*
SET_FREE_ADDRESS
COPYAPPEND copy service
DST.TS_SERVICE
SRC.TS_SERVICE
COPYAPPEND copy address to free address
FORCE (even if empty)
DST.TS_FREE_ADDR
SRC.TS_NUMBER

```

Section TO_MAIL_RECIP

- Destination address must be SET_FREE_ADDRESS.
- DST.TS_SERVICE must be NOTES.
- DST.TS_FREE_ADDR must be the address of the NOTES recipient

```

REM -----
TO_MAIL_RECIP recipient of message to NOTES
REM -----
REM -----
REM only 1 entry: service = NOTES, address = destination address
REM -----
ENTRY
*
*

```

```

NOTES    service = NOTES
SET FREE ADDRESS
COPYAPPEND copy address (without service prefix)
DST.TS_FREE_ADDR
SRC.TS_TRUENUMBER
    
```

The TC/LINK-SM Standard MAPfile, Fully Documented

- MAPFile name: SMTP.MAP
- The complete address is passed in TS_FREE_ADDR

Section TO_TC_ORIG

- Source address is type SET_FREE_ADDRESS
- Complete address information is in SRC.TS_FREE_ADDR

```

REM *****
REM 1.07.02 Added feature for SMTP originator address from Userprofile
REM 1.07.04 KCS userids with blanks put to ""
REM 1.08.01 Added section for SAPConnect Link ("SMTP:")
REM 1.09.00 Enhanced originator mapping: proxy address is full SMTP A.
REM *****
REM -----
TO_TC_ORIG  originator of message to KCS
REM -----
REM -----
REM originator with KCS shadow user (enhanced originator mapping)
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET FREE ADDRESS
SEARCH_USER_ADDRESS look for KCS user with mail address
SRC.TS_FREE_ADDR
""
REM or ev. REG.TOPCALL\ServiceTC ?
REM -----
REM originator with shadow user (on link domain, via user id)
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET FREE ADDRESS
COPYFMT  separate address components
SRC.TS_FREE_ADDR
"@???!!"
DST.TS_CORREL_1
DST.TS_CORREL_1
DST.TS_CORREL_1
DST.TS_RECP_ID  find user id ...
DST.TS_CORREL_2  ... and domain
COMPARESTRING  check if user is on link domain
IEQU
DST.TS_CORREL_2
REG.TCLSM\SMLinkDomain
SEARCH_USER_ID  search KCS user with this ID
ALL
    
```

```

REM -----
REM originator without shadow user: keep Free address
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND copy address
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address

```

Section TO_TC_RECIP

- Source address is SET_FREE_ADDRESS
- Complete address (including service) is in SRC.TS_FREE_ADDR
- Different syntax possibilities

```

REM -----
TO_TC_RECIP recipient of message to KCS
REM -----
REM -----
REM recipient on other domain (not link domain):
REM will go out again via TC/LINK-SM
REM -----
ENTRY ----- check if it is for me! If not: add link queue, out again!
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT separate address components
SRC.TS_FREE_ADDR (isolate domain)
"@???!!"
DST.TS_CORREL_1
DST.TS_CORREL_1
DST.TS_CORREL_1
DST.TS_RECIP_ID
DST.TS_CORREL_2 domain
COMPARESTRING proceed only if domain is not link domain
INEQU
DST.TS_CORREL_2
REG.TCLSM\SMLinkDomain
COPYAPPEND copy address
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address
REM -----
REM address on link domain, syntax "fullname#service#number#answerback"
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
ISOLATE_LOCALPART get local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
COPYFMT separate address components
DST.TS_FREE_ADDR
"#?!!!" fullname may be missing
DST.TS_FULLNAME

```

```

DST.TS_SERVICE
DST.TS_CORREL_1
DST.TS_CORREL_2
COPYAPPEND concatenate number and answerback
DST.TS_CORREL_1
DST.TS_CORREL_1
"#"
DST.TS_CORREL_2
SEARCH_ADDRTYPE build address from service, number and answerback
DST.TS_SERVICE
DST.TS_CORREL_1
SEP=#
REM -----
REM address on link domain, syntax "fullname#service#number"
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
ISOLATE_LOCALPART get local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
COPYFMT separate address components
DST.TS_FREE_ADDR
"#!!!"
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_CORREL_1
SEARCH_ADDRTYPE build address
DST.TS_SERVICE
DST.TS_CORREL_1
SEP=#
REM -----
REM address on link domain, syntax "service#number"
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
ISOLATE_LOCALPART get local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
COPYFMT isolate address components
DST.TS_FREE_ADDR
"#!!"
DST.TS_SERVICE
DST.TS_CORREL_1
SEARCH_ADDRTYPE build address
DST.TS_SERVICE
DST.TS_CORREL_1
SEP=#
REM -----
REM address on link domain, single parameter (KCS UserId)
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
ISOLATE_LOCALPART get local part of SMTP address ...
SRC.TS_FREE_ADDR
DST.TS_RECPI_ID ... into userid

```

```

SEARCH_USER_ID  search user or addressbook entry with this ID
ALL
REM -----
REM no match till now, syntax "fullname#userid#node" via default TC service
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
ISOLATE_LOCALPART get local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
COPYFMT  separate components of local part
DST.TS_FREE_ADDR
"#?!?"
DST.TS_FULLNAME fullname
DST.TS_CORREL_1  userid
DST.TS_CORREL_2  node
COPYAPPEND  service = TOPCALL
DST.TS_SERVICE
"TOPCALL"
CVTSERVICE  build a KCS address
DST.TS_SERVICE
DST.TS_CORREL_1
DST.TS_CORREL_2
SEARCH_MISSING SVC must be done, adjusts address type if SMTPIN service

```

Section TO_MAIL_ORIG

- Destination address must be SET_FREE_ADDRESS.
- Complete address must be in DST.TS_FREE_ADDR.

```

REM -----
REM  originator of message to SMTP
REM -----
REM -----
REM originator has SMTP address (via our link): take only this address
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
CHECKLINKQUEUE proceed only if address via this TC/LINK-SM
COPYAPPEND  take SMTP address
DST.TS_FREE_ADDR
SRC.TS_TRUENUMBER
ONLYTHIS  ignore all other addresses of the person
REM -----
REM originator is KCS user, UserID contains blanks: "UserID"@domain
REM -----
ENTRY
*
SET_TC_ADDRESS  KCS users only
*
SET_FREE_ADDRESS
COPYFMT  check if userid contains blank
SRC.TS_TC_USERID
"!!"
DST.TS_CORREL_1
DST.TS_CORREL_2

```



```
COPYAPPEND build SMTP address "UserID"@LinkDomain
DST.TS_FREE_ADDR
"
SRC.TS_TC_USERID
"
"@
REG.TCLSM\SMLinkDomain LinkDomain from configuration
REM -----
REM originator is KCS user, UserID without blanks: UserID@domain
REM -----
ENTRY
*
SET_TC_ADDRESS KCS users only
*
SET_FREE_ADDRESS
COPYAPPEND build SMTP address UserID@LinkDomain
DST.TS_FREE_ADDR
SRC.TS_TC_USERID
"@
REG.TCLSM\SMLinkDomain LinkDomain from configuration
REM -----
REM Fax address: fullname#service#number#answerback@domain
REM -----
ENTRY
*
SET_FAX_ADDRESS Fax addresses only
*
SET_FREE_ADDRESS
COPYAPPEND build address
DST.TS_FREE_ADDR
SRC.TS_FULLNAME
"#
SRC.TS_SERVICE
"#
SRC.TS_FAX_NUMBER
"#
SRC.TS_FAX_ANSWERB
"@
REG.TCLSM\SMLinkDomain Linkdomain from configuration
REM -----
REM Telex address: fullname#service#number#answerback@LinkDomain
REM -----
ENTRY
*
SET_TX_ADDRESS Telex addresses only
*
SET_FREE_ADDRESS
COPYAPPEND build address
DST.TS_FREE_ADDR
SRC.TS_FULLNAME
"#
SRC.TS_SERVICE
"#
SRC.TS_TX_NUMBER
"#
SRC.TS_TX_ANSWERB
"@
REG.TCLSM\SMLinkDomain LinkDomain from configuration
REM -----
REM Teletex address: fullname#service#number#answerback@LinkDomain
REM -----
ENTRY
*
SET_TTX_ADDRESS Teletex addresses only
```

```

*
SET FREE_ADDRESS
COPYAPPEND build address
DST.TS_FREE_ADDR
SRC.TS_FULLNAME
"#"
SRC.TS_SERVICE
"#"
SRC.TS_TTX_NUMBER
"#"
SRC.TS_TTX_ANSWERB
"@
REG.TCLSM\SMLinkDomain LinkDomain from configuration
REM -----
REM SAPConnect SMTP address (in syntax SMTP:user@domain)
REM This should work for SMTP addresses from Exchange as well
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT proceed only if number contains `:`
SRC.TS_TRUENUMBER
":!!"
VAR.0
VAR.1
COMPARESTRING .. and if first part is "SMTP:"
IEQU
VAR.0
"SMTP"
COPYAPPEND take only real SMTP address (without "SMTP:")
DST.TS_FREE_ADDR
VAR.1
REM -----
REM Other type of free address, syntax "fullname#service#number"@LinkDomain
REM -----
ENTRY
*
SET_FREE_ADDRESS Free addresses only
*
SET_FREE_ADDRESS
COPYAPPEND build address
DST.TS_FREE_ADDR
" address may contain blank, therefore in " "
SRC.TS_FULLNAME
"#"
SRC.TS_SERVICE
"#"
SRC.TS_NUMBER
"
"@
REG.TCLSM\SMLinkDomain LinkDomain from configuration
REM -----
REM X400 address, syntax "service#x400number"@LinkDomain
REM -----
ENTRY
*
SET_X400_ADDRESS X400 addresses only
*
SET_FREE_ADDRESS
COPYAPPEND build address
DST.TS_FREE_ADDR
"

```

```
SRC.TS_SERVICE
"#
SRC.TS_NUMBER
"
"@
REG.TCLSM\SMLinkDomain LinkDomain from configuration
```

Section TO_MAIL_RECIP

- Destination address type must be SET_FREE_ADDR.
- Complete address must be in DST.TS_FREE_ADDR.

```
REM -----
TO_MAIL_RECIP  recipient of message to SMTP
REM -----
REM -----
REM all recipients are addressed as FREE address type
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND  just copy address
DST.TS_FREE_ADDR
SRC.TS_TRUENUMBER
```

The TC/LINK-SJ Standard MAPfile, Fully Documented

- MAPFile name: SJ.MAP

Section TO_TC_ORIG

- Source address is of type SET_FREE_ADDRESS.
- The complete address is in SRC.TS_FREE_ADDR.

```
REM *****
REM 1.09.01 Initial Release for HP ScanJet 5
REM *****
REM -----
TO_TC_ORIG  originator of message to KCS
REM -----
REM -----
REM Anonymous -> Guest user as originator
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COMPARESTRING  check if anonymous originator
IEQU
SRC.TS_FREE_ADDR
"HP_SCANNER-Guest"
COPYAPPEND  originator = guest user
DST.TS_RECIP_ID
REG.TOPCALL\GuestUser
SEARCH_USER_ID  get guest user entry from KCS
```

```

REM -----
REM originator with shadow user (ID includes the "HP_SCANNER-" prefix)
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND UserID = source address
DST.TS_RECIP_ID
SRC.TS_FREE_ADDR
SEARCH_USER_ID search KCS user with this ID
REM -----
REM originator with shadow user (ID without the "HP_SCANNER-" prefix)
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT check if address includes '-'
SRC.TS_FREE_ADDR
"-!!" and remove all before '-'
DST.TS_CORREL_1
DST.TS_RECIP_ID the rest is the UserID
SEARCH_USER_ID search KCS user with this ID
REM -----
REM originator with shadow user (enhanced originator mapping)
REM shadow user has an SJ proxy address (without "HP_SCANNER-" prefix)
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT check if address includes '-'
SRC.TS_FREE_ADDR
"-!!" and remove all before '-'
DST.TS_CORREL_1
DST.TS_RECIP_ID the rest is the proxy address
SEARCH_USER_ADDRESS search KCS user with this proxy address
DST.TS_RECIP_ID
REM -----
REM originator without shadow user: keep Free address
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND just copy address
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address

```

Section TO_TC_RECIP

- Source address is of type SET_FREE_ADDRESS.
- Complete address is in SRC.TS_FREE_ADDR.

```

REM -----
TO_TC_RECIP recipient of message to KCS

```

```

REM -----
REM -----
REM 3 or 4 parameters, syntax: fullname,service,number,answerback
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT separate address components
SRC.TS_FREE_ADDR
",?!!!"
DST.TS_FULLNAME fullname is optional
DST.TS_SERVICE
DST.TS_CORREL_1
DST.TS_CORREL_2
COPYAPPEND concatenate number and answerback
DST.TS_CORREL_1
DST.TS_CORREL_1
", "
DST.TS_CORREL_2
SEARCH_ADDRTYPE build address
DST.TS_SERVICE
DST.TS_CORREL_1
SEP=,
REM -----
REM 3 parameters, syntax: fullname,service,number
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT separate address components
SRC.TS_FREE_ADDR
",!!!"
DST.TS_FULLNAME
DST.TS_SERVICE
DST.TS_CORREL_1
SEARCH_ADDRTYPE build address
DST.TS_SERVICE
DST.TS_CORREL_1
SEP=,
REM -----
REM 2 parameters, addressbook entry
REM -----
ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT separate address components
SRC.TS_FREE_ADDR
",!!"
DST.TS_SERVICE
DST.TS_RECIP_ID second parameter is ID
SEARCH_USER_ID search user or addressbook entry with this ID
ALL
REM -----
REM 2 parameters, syntax: service,number
REM -----
ENTRY
*
SET_FREE_ADDRESS

```

```

*
SET FREE_ADDRESS
COPYFMT separate address components
SRC.TS_FREE_ADDR
",!!"
DST.TS_SERVICE
DST.TS_CORREL_1
SEARCH_ADDRTYPE build address
DST.TS_SERVICE
DST.TS_CORREL_1
SEP=,
REM -----
REM single parameter, addressbook entry
REM -----
ENTRY
*
SET FREE_ADDRESS
*
SET FREE_ADDRESS
COPYAPPEND ID = complete address
DST.TS_RECIP_ID
SRC.TS_FREE_ADDR
SEARCH_USER_ID search user or addressbook entry with this ID
ALL
REM -----
REM single parameter, not in addressbook: use default service
REM -----
ENTRY
*
SET FREE_ADDRESS
*
SET FREE_ADDRESS
SEARCH_ADDRTYPE build address
REG.Options\DefaultRecipientService default service from configuration
SRC.TS_FREE_ADDR address
SEP=,
REM -----
REM address still cannot be mapped, take it "as is"
REM nondel will not work !
REM -----
ENTRY
*
SET FREE_ADDRESS
*
SET FREE_ADDRESS
COPYAPPEND just copy address
DST.TS_FREE_ADDR
SRC.TS_TRUENUMBER

```

Section TO_MAIL_ORIG

- Dummy section, never used (Scanjet does not receive messages)

```

REM -----
TO_MAIL_ORIG originator of message to SCANJET
REM -----
REM -----
REM there is no message to SCANJET, this dummy entry is not used !
REM -----
ENTRY
*
*
*

```



```

SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND build fullname for cover variable
DST.TS_FULLNAME (GivenName SurName)
DST.TS_X400_GNAME
" "
DST.TS_X400_SNAME
COPYAPPEND company = organization (for cover)
DST.TS_COMPANY
DST.TS_X400_ORG
COPYAPPEND department = OUI (for cover)
DST.TS_DEPTM
DST.TS_X400_OUI
UPDATE_X400TEXT build X.400 string (from X.400 fields)
SEARCH_LINKSVC add link queue (or service for it) to address

```

Section TO_TC_RECIP

- Source address is always SET_X400_ADDRESS.
- The X400 Address holds the complete addressing information.
(in the standard X400 fields or in DDA values)
- DST.TS_CORREL_2 must hold the per-recipient correlation information ("X4NOTIF>" followed by the original X.400 address string)

```

REM -----
TO_TC_RECIP recipient of message to KCS
REM -----
REM -----
REM number/answerback and service specified in DDA value (DDA1)
REM take cover variables from other X.400 fields
REM (only if "GivenName SurName" != " ")
REM -----
ENTRY
*
SET_X400_ADDRESS
*
*
COPY copy address to destination
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT update X.400 address string from X.400 fields
COPYAPPEND save complete X.400 address string in VAR.0
VAR.0 (used later for correlation)
DST.TS_X400_ADDR
COPYAPPEND take service from DDA type 1
DST.TS_SERVICE
SRC.TS_D1NAME
COPYAPPEND fullname = "GivenName SurName"
DST.TS_FULLNAME
SRC.TS_X400_GNAME
" "
SRC.TS_X400_SNAME
COMPARESTRING proceed only if fullname holds information
NEQU
DST.TS_FULLNAME
" "
COPYAPPEND company = organization (for cover variable)
DST.TS_COMPANY
SRC.TS_X400_ORG
COPYAPPEND department = OUI
DST.TS_DEPTM

```

```

SRC.TS_X400_OU1
COPYFMT separate address components (DDA value 1)
SRC.TS_D1DATA
",!???"
VAR.3 number
VAR.1 answerback (optional)
DST.TS_SALUTE salutation (optional)
DST.TS_FREETEXT free text (optional)
COPYAPPEND concatenate number and answerback
VAR.2
VAR.3
", "
VAR.1
SEARCH_ADDRTYPE build address
DST.TS_SERVICE
VAR.2
SEPALL=, remove \,' if at end
COPYAPPEND shortname = "GivenName SurName"
DST.TS_RECIP_ID
SRC.TS_X400_GNAME
" "
SRC.TS_X400_SNAME
COPYAPPEND store original X.400 address
DST.TS_CORREL_2 in per-recipient correlation (for notif)
"X4NOTIF">
VAR.0
REM -----
REM number/answerback and service specified in DDA value (DDA1)
REM take cover variables also from DDA value
REM (only if "GivenName SurName" = " ")
REM -----
ENTRY
*
SET_X400_ADDRESS
*
*
COPY copy address (must be in dst for next rule)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT update X.400 address string according to fields
COPYAPPEND save X.400 address string in VAR.0
VAR.0 (used later for correlation info)
DST.TS_X400_ADDR
COPYAPPEND service = DDA type (DDA1)
DST.TS_SERVICE
SRC.TS_D1NAME
COPYAPPEND write empty string to fullname
FORCE (we need it for notification)
DST.TS_FULLNAME
""
COPYFMT take address and cover variables from DDA value
SRC.TS_D1DATA
",?!?????"
DST.TS_FULLNAME fullname (optional)
VAR.3 number
VAR.1 answerback (opt.)
DST.TS_COMPANY company (opt.)
DST.TS_DEPTM department (opt.)
DST.TS_SALUTE salutation (opt.)
DST.TS_FREETEXT free text (opt.)
COPYAPPEND concatenated number and answerback
VAR.2
VAR.3
", "

```

```

VAR.1
SEARCH ADDRTYPE build address
DST.TS_SERVICE
VAR.2
SEPALL=, (remove trailing comma)
COPYAPPEND shortname = "GivenName Surname"
DST.TS_RECIP_ID
SRC.TS_X400_GNAME
" "
SRC.TS_X400_SNAME
COPYAPPEND store original X.400 address string
DST.TS_CORREL_2 as per-recipient correlation for notif.
"X4NOTIF>"
VAR.0
REM -----
REM recipient is KCS user (DDA not used)
REM UserID = "GivenName SurName"
REM -----
ENTRY
*
SET_X400_ADDRESS
*
SET_TC_ADDRESS
COPY copy address (must be in dst for next rule)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT write X.400 address string from fields
COPYAPPEND save X.400 address string in VAR.0
VAR.0 (later used as correlation)
DST.TS_X400_ADDR
COPYAPPEND UserID = "GivenName SurName"
DST.TS_RECIP_ID
SRC.TS_X400_GNAME
" "
SRC.TS_X400_SNAME
COPYAPPEND store original X.400 address string as correlation
SRC.TS_CORREL_2 (for notification)
"X4NOTIF>"
VAR.0
SEARCH_USER_ID search KCS user or addressbook entry with this ID
ALL
REM -----
REM recipient is KCS user (DDA values not used)
REM UserID = SurName
REM -----
ENTRY
*
SET_X400_ADDRESS
*
SET_TC_ADDRESS
COPY copy address (must be in dst for next rule)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT build X.400 address string
COPYAPPEND save X.400 address string in VAR.0
VAR.0 for later use as correlation info.
DST.TS_X400_ADDR
COPYAPPEND UserID = SurName
DST.TS_RECIP_ID
SRC.TS_X400_SNAME
COPYAPPEND store original X.400 address string an correlation
SRC.TS_CORREL_2 (for notification)
"X4NOTIF>"
VAR.0

```

```

SEARCH_USER_ID  search user or addressbook entry with this ID
ALL
REM -----
REM default (no DDA values, no KCS user)
REM take X.400 address as it is and add default X.400 service
REM -----
ENTRY
*
SET_X400_ADDRESS
*
SET_X400_ADDRESS
COPY    copy address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT write X.400 address string acc. To X.400 fields
COPYAPPEND fullname = "GivenName SurName" (for cover)
DST.TS_FULLNAME
DST.TS_X400_GNAME
" "
DST.TS_X400_SNAME
COPYAPPEND company = Organization (for cover)
DST.TS_COMPANY
DST.TS_X400_ORG
COPYAPPEND department = OU1 (for cover)
DST.TS_DEPTM
DST.TS_X400_OU1
SEARCH_MISSINGSVC add default X.400 service

```

Section TO_MAIL_ORIG

- Destination addresses must be of type SET_X400_ADDRESS.
- The individual fields of the destination X.400 address must be used to hold the addressing information.
- If SRC.TS_CORREL_2 starts with "X4NOTIF>", this is the originator of a notification to TC/LINK-X4. Then take the X.400 address from SRC.TS_CORREL_2.
- If the source address is already an X.400 address and contains no correlation from TC/LINK-X4, DST.TS_SERVICE must be X400ORIG.

```

REM -----
TO_MAIL_ORIG  originator of message to X.400
REM -----
REM -----
REM originator has X.400 address
REM mark this address with service X400ORIG.
REM Special part of the link checks for service X400ORIG and
REM takes this address preferably for the originator
REM -----
ENTRY
*
SET_X400_ADDRESS only for X.400 addresses
X400ORIG service becomes X400ORIG
SET_X400_ADDRESS
COPY    just copy address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
REM -----
REM originator address is not X.400, notification for msg from X.400
REM (complete X.400 address is in correlation)
REM -----
ENTRY
*
*

```

```

EMAIL    service becomes EMAIL
SET_X400_ADDRESS
COPYFMT  only proceed if this is originator of a notification
SRC.TS_CORREL_2  (correlation marker)
">!!"
VAR.1
VAR.0
COMPARESTRING  check the correlation marker
EQU
VAR.1
"X4NOTIF"  (must be X4NOTIF)
COPYAPPEND  take address string from correlation
DST.TS_X400_ADDR
VAR.0
UPDATE_X400FIELDS  update X.400 fields accordingly
REM -----
REM originator has TC address
REM -----
ENTRY
*
SET_TC_ADDRESS  TC addresses only
EMAIL    service becomes EMAIL
SET_X400_ADDRESS
COPYFMT  take SurName (and GivenName) from User ID
SRC.TS_TC_USERID
" ?!"
DST.TS_X400_GNAME
DST.TS_X400_SNAME
REM -----
REM originator address is SAPCONNECT X400 address
REM -----
ENTRY
*
SET_FREE_ADDRESS  Free addresses only
*  no service
SET_X400_ADDRESS
COPYFMT  check if number contains `:' ....
SRC.TS_TRUENUMBER
":!!"
VAR.0
VAR.1
COMPARESTRING  and if first part is "X400:"
IEQU
VAR.0
"X400"
COPYAPPEND  take X.400 address string (second part of address)
DST.TS_X400_ADDR
VAR.1
UPDATE_X400FIELDS  and update X.400 fields accordingly
REM -----
REM any other address: put service and number into DDA1
REM -----
ENTRY
*
*
EMAIL    service becomes EMAIL
SET_X400_ADDRESS
COPYAPPEND  DDA1 Name = service
DST.TS_D1NAME
SRC.TS_SERVICE
COPYAPPEND  DDA1 Value = number
DST.TS_D1DATA
SRC.TS_NUMBER

```



```

*
SET TC_ADDRESS
COPY copy address (must be in dst for next rule)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE X400FIELDS build X.400 fields from address string
COPYAPPEND UserID = "GivenName SurName"
DST.TS_RECIP_ID
DST.TS_X400_GNAME
" "
DST.TS_X400_SNAME
SEARCH_USER_ID search for user with this UserID
REM -----
REM originator with shadow user, UserID = SurName
REM originator address is X400
REM -----
ENTRY
X40 only if service is X40
SET_X400_ADDRESS only if address type is X400
*
SET TC_ADDRESS
COPY copy address (must be in dst for next rule)
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400FIELDS build X.400 address fields from address string
COPYAPPEND UserID = SurName
DST.TS_RECIP_ID
DST.TS_X400_SNAME
SEARCH_USER_ID search for user with this UserID
REM -----
REM originator without shadow user, SMTP address
REM -----
ENTRY
INT only if service is INT
SET_FREE_ADDRESS only if address type is Free
*
SET_FREE_ADDRESS
COPYAPPEND build address string:
DST.TS_FREE_ADDR "SMTP:SmtAddress"
"SMTP"
":"
SRC.TS_FREE_ADDR address type becomes Free
SEARCH_LINKSVC add link queue (or service for it) to address
REM -----
REM originator without shadow user, X400 addresses
REM -----
ENTRY
X40 only if service is X40
SET_X400_ADDRESS only if address type is X400
*
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND build address string:
DST.TS_FREE_ADDR "X400:X400Address"
"X400"
":"
SRC.TS_X400_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address
REM -----
REM originator with shadow user, enhanced originator mapping
REM originator address is FAX address
REM -----
ENTRY
FAX only if service is FAX
SET_FREE_ADDRESS only if address type is Free

```



```

*
SET FREE ADDRESS
SEARCH_USER_ADDRESS search user with this Fax proxy address
SRC.TS_FREE_ADDR
""
REM or REG.Setup\ServiceSCFAX\Name
REG.SAP\SCLinkDIDLen (only configured number of trailing digits = DID)
REM -----
REM originator without shadow user, FAX address
REM -----
ENTRY
FAX only if service is FAX
SET_FREE_ADDRESS only if address type is Free
*
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND build address:
DST.TS_FREE_ADDR "FAX:FaxAddress"
"FAX"
": "
SRC.TS_FREE_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address

```

Section TO_TC_RECIP

- Three types of source addresses: SMTP syntax, X400 syntax, FAX syntax.
- SMTP syntax: source address type is SET_FREE_ADDRESS, SRC.TS_SERVICE is "INT"
- X400 syntax: source address type is SET_X400_ADDRESS, SRC.TS_SERVICE is "X40". The correct X400 address is in SRC.TS_X400_ADDR.
- FAX syntax: source address type is SET_FREE_ADDRESS, SRC.TS_SERVICE is "FAX"
- DST.TS_CORREL_2 must hold the per-recipient correlation information ("SCNOTIF>" followed by the original address string)

```

REM -----
TO_TC_RECIP recipient of message to KCS
REM -----
REM -----
REM Recipient has Fax address (service = "FAX", Free address = number)
REM -----
ENTRY
FAX only if service is FAX
SET_FREE_ADDRESS
*
SET_FAX_ADDRESS address type becomes Fax
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)
SRC.TS_SERVICE
": "
SRC.TS_FREE_ADDR
COPYAPPEND fax number = specified address
DST.TS_FAX_NUMBER
SRC.TS_FREE_ADDR
COPYAPPEND shortname = fax number (for cover variables)
DST.TS_RECIP_ID
DST.TS_FAX_NUMBER
SEARCH_MISSINGSVC service = default Fax service (from configuration)
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
REM -----

```

```

REM Recipient has X400 address (but not own X400 domain)
REM (service = "X40", X400 address)
REM is sent to default X400 service
REM -----
ENTRY
X40    only if service is "X40"
SET_X400_ADDRESS only if address type is X400
*
SET_X400_ADDRESS address type becomes X400
COPY    just copy the X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND store original address (including service) in VAR.0
VAR.0    (used later as per-recipient correlation)
SRC.TS_SERVICE
":"
DST.TS_X400_ADDR
UPDATE_X400FIELDS update the X400 address fields
REM now check if address is in our own domain
COPYAPPEND build address string for own domain (in VAR.1)
VAR.1
REG.SAP\SCLinkX400Country take address components from configuration
REG.SAP\SCLinkX400ADMD
REG.SAP\SCLinkX400PRMD
REG.SAP\SCLinkX400Org
REG.SAP\SCLinkX400OU1
COPYAPPEND build domain address of this user (in VAR.2)
VAR.2
DST.TS_X400_C
DST.TS_X400_AD
DST.TS_X400_PD
DST.TS_X400_ORG
DST.TS_X400_OU1
REM    check if both addresses are equal
COMPARESTRING proceed only if recipient address is not in own domain
INEQU
VAR.1
VAR.2
SEARCH_MISSINGSVC take default X400 service
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
REM -----
REM Recipient has X400 address, service and number in DDA
REM -----
ENTRY
X40    only if service is "X40"
SET_X400_ADDRESS only if address type is X400
*
*
COPY    just copy the X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND store original address (including service) in VAR.0
VAR.0    (used later as per-recipient correlation)
SRC.TS_SERVICE
":"
DST.TS_X400_ADDR
UPDATE_X400FIELDS update the X.400 address fields (from string)
COPYAPPEND service = DDA1 name
DST.TS_SERVICE
DST.TS_D1NAME
COPYFMT take number and cover vars from DDA1 value

```

```

DST.TS_D1DATA
",!?????" syntax:number,answerback,company,department,salutation,freetext
VAR.3
VAR.1
DST.TS_COMPANY
DST.TS_DEPTM
DST.TS_SALUTE
DST.TS_FREETEXT
COPYAPPEND concatenate number and answerback
VAR.2
VAR.3
", "
VAR.1
SEARCH_ADDRTYPE build address from service and number+answerback
DST.TS_SERVICE
VAR.2
SEPALL=,
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
REM -----
REM recipient has X400 address and has TC shadow user with
REM TC UserID = "GivenName SurName"
REM -----
ENTRY
X40 only if service is "X40"
SET_X400_ADDRESS only if address type is X400
*
SET_TC_ADDRESS (this is changed later by SEARCH_USER_ID)
COPY copy X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)
SRC.TS_SERVICE
": "
DST.TS_X400_ADDR
UPDATE_X400FIELDS update X400 address fields (acc to string)
COPYAPPEND TC UserID = "GivenName SurName"
DST.TS_RECIP_ID
DST.TS_X400_GNAME
" "
DST.TS_X400_SNAME
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
SEARCH_USER_ID search for KCS user or recipient with this UserID
ALL
REM -----
REM Recipient has X400 address, is KCS user with
REM UserID = SurName
REM -----
ENTRY
X40 only if service is "X40"
SET_X400_ADDRESS only if address type is X400
*
SET_TC_ADDRESS (this is changed later by SEARCH_USER_ID)
COPY copy the X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)

```

```

SRC.TS_SERVICE
":"
DST.TS_X400_ADDR
UPDATE_X400FIELDS update X400 address fields (acc to string)
COPYAPPEND UserID = SurName
DST.TS_RECIP_ID
DST.TS_X400_SNAME
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
SEARCH_USER_ID search KCS user or recipient with this UserID
ALL
REM -----
REM recipient has SMTP address (not in our domain)
REM -----
ENTRY
INT only if service is "INT"
SET_FREE_ADDRESS and address type is Free
*
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)
SRC.TS_SERVICE
":"
SRC.TS_FREE_ADDR
COPYFMT separate local part and domain
SRC.TS_FREE_ADDR
"@!!"
VAR.1
VAR.2
COMPARESTRING check if domain is ours, only proceed if different
INEQU
VAR.2
REG.SAP\SCLinkSMTPDomain
COPYAPPEND destination address = whole SMTP address
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
COPYAPPEND service = default SMTP service (from config)
DST.TS_SERVICE
REG.TOPCALL\SMTPService
SEARCH_ADDRTYPE build address from service and SMTP address
DST.TS_SERVICE
DST.TS_FREE_ADDR
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
REM -----
REM recipient has SMTP address, local part = "service#number#answerback"
REM -----
ENTRY
INT only if service is "INT"
SET_FREE_ADDRESS and address type is Free
*
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)
SRC.TS_SERVICE
":"
SRC.TS_FREE_ADDR
ISOLATE_LOCALPART isolate local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR

```

```

COPYFMT separate components of local part
DST.TS_FREE_ADDR (service#number#answerback)
"#!!"
DST.TS_SERVICE
VAR.1
VAR.2
COPYAPPEND concatenate number and answerback
VAR.1
VAR.1
"#"
VAR.2
SEARCH ADDRTYPE build address from service, number and answerback
DST.TS_SERVICE
VAR.1
SEP=#
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
REM -----
REM recipient has SMTP address, local part is "service#number"
REM -----
ENTRY
INT only if service is "INT"
SET_FREE_ADDRESS and address type is Free
*
SET_FREE_ADDRESS
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)
SRC.TS_SERVICE
":"
SRC.TS_FREE_ADDR
ISOLATE LOCALPART separate local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
COPYFMT separate components of local part:
DST.TS_FREE_ADDR
"#!!" service#number
DST.TS_SERVICE
VAR.1
SEARCH ADDRTYPE build address from service and number
DST.TS_SERVICE
VAR.1
SEP=#
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0
REM -----
REM recipient has SMTP address, local part is TC UserId
REM -----
ENTRY
INT only if service is "INT"
SET_FREE_ADDRESS and address type is Free
*
SET_FREE_ADDRESS
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)
SRC.TS_SERVICE
":"
SRC.TS_FREE_ADDR
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"

```

```

VAR.0
ISOLATE LOCALPART separate local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_RECP_ID = KCS UserID
SEARCH_USER_ID search KCS user or recipient with this ID
ALL
REM -----
REM recipient has SMTP address, is KCS user on a remote KCS
REM local part is "UserID#Node"
REM -----
ENTRY
INT only if service is "INT"
SET_FREE_ADDRESS and address type is Free
*
SET_FREE_ADDRESS
COPYAPPEND store original address (including service) in VAR.0
VAR.0 (used later as per-recipient correlation)
SRC.TS_SERVICE
":"
SRC.TS_FREE_ADDR
ISOLATE LOCALPART separate local part of SMTP address
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
COPYFMT separate components of local part
DST.TS_FREE_ADDR
"#!?" UserID#Node
VAR.1
VAR.2
COPYAPPEND prepare variable with keyword TOPCALL for CVTSERVICE
DST.TS_SERVICE
"TOPCALL"
CVTSERVICE build an address with default TC service and
DST.TS_SERVICE UserID and Node
VAR.1
VAR.2
SEARCH_MISSINGSVC once again to be sure: take default TC service
REM (this deals with Free service SMTPIN)
COPYAPPEND store original number as correlation (returned in notif)
DST.TS_CORREL_2
"SCNOTIF>"
VAR.0

```

Section TO_MAIL_ORIG

- Destination address type must always be SET_FREE_ADDRESS.
- For originator of notification to SAP, take the address stored in SRC.TS_CORREL_2.
- For other originators, use SMTP, X400 or FAX syntax, according to the source address type.
- SMTP syntax: DST.TS_SERVICE is "INT", DST.TS_FREE_ADDR contains address. If the message comes from a service other than "SMTP", use the configured link domain.
- X400 syntax: DST.TS_SERVICE is "X40", DST.TS_FREE_ADDR is the X.400 address string. This syntax is only used for X400 source addresses.
- FAX syntax: DST.TS_SERVICE is "FAX", DST.TS_FREE_ADDR is the fax number. This syntax is only used for source addresses of type Fax.

```

REM -----
TO_MAIL_ORIG originator of message to SAPCONNECT
REM -----
REM -----
REM originator of notification to SAP: take address from SRC.TS_CORREL_2

```

```
REM -----
ENTRY
*
*
*
SET_FREE_ADDRESS address type becomes Free
COPYFMT check if correlation field exists
SRC.TS_CORREL_2
">!!"
VAR.1
VAR.0
COMPARESTRING check if original message was from TC/LINK-SC
EQU
VAR.1
"SCNOTIF"
COPYFMT take service and Free address from correlation
VAR.0
":!!"
DST.TS_SERVICE
DST.TS_FREE_ADDR
REM -----
REM originator has TC address: use SMTP syntax (UserID@LinkDomain)
REM -----
ENTRY
*
SET_TC_ADDRESS only if address type is TC (KCS internal)
INT service becomes "INT"
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND build address string: UserID@LinkDomain
DST.TS_FREE_ADDR
SRC.TS_TC_USERID
"@
REG.SAP\SCLinkSMTPDomain take LinkDomain from configuration
REM -----
REM originator has Fax address, use Fax syntax
REM -----
ENTRY
*
SET_FAX_ADDRESS only if address type is Fax
FAX service becomes "FAX"
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND address string: FaxNumber
DST.TS_FREE_ADDR
SRC.TS_FAX_NUMBER
REM -----
REM originator has Telex address: use SMTP syntax
REM service#number#answerback@LinkDomain
REM -----
ENTRY
*
SET_TX_ADDRESS only if address type is Telex
INT service becomes "INT"
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND build address string:
DST.TS_FREE_ADDR service#number#answerback@LinkDomain
SRC.TS_SERVICE
"#
SRC.TS_TX_NUMBER
"#
SRC.TS_TX_ANSWERB
"@
REG.SAP\SCLinkSMTPDomain LinkDomain from configuration
REM -----
REM originator has Teletex address: use SMTP syntax
```

```

REM service#number#answerback@LinkDomain
REM -----
ENTRY
*
SET_TTX_ADDRESS only if address type is Teletex
INT service becomes "INT"
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND build address string:
DST.TS_FREE_ADDR service#number#answerback@LinkDomain
SRC.TS_SERVICE
"#"
SRC.TS_TTX_NUMBER
"#"
SRC.TS_TTX_ANSWERB
"@
REG.SAP\SCLinkSMTPDomain LinkDomain from configuration
REM -----
REM originator has SMTP address (service SMTP): use SMTP syntax
REM and leave address as it is
REM -----
ENTRY
SMTP only if service is "SMTP"
SET_FREE_ADDRESS and if address type is Free
INT service becomes "INT"
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND just copy the address string
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
REM -----
REM originator has Free address: use SMTP syntax
REM service#number@LinkDomain
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
INT service becomes "INT"
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND build address string:
DST.TS_FREE_ADDR service#number@LinkDomain
SRC.TS_SERVICE
"#"
SRC.TS_NUMBER
"@
REG.SAP\SCLinkSMTPDomain LinkDomain from configuration
REM -----
REM originator has X400 address, use X400 syntax
REM -----
ENTRY
*
SET_X400_ADDRESS only if address type is X400
X40 service becomes "X40"
SET_FREE_ADDRESS address type becomes Free
COPYAPPEND copy X400 address string
DST.TS_FREE_ADDR
SRC.TS_X400_ADDR

```

Section TO_MAIL_RECIP

- Destination address type must be SET_FREE_ADDRESS.
- DST.TS_SERVICE must be one of "INT", "X40" and "FAX".

- Mapfile entries assume that the original address string contains a prefix indicating the address syntax: "SMTP:", "X400:" or "FAX:".

```

REM -----
TO MAIL_RECP    recipient of message to SAPCONNECT
REM -----
REM -----
REM recipient address is in SMTP syntax
REM -----
ENTRY
*
SET_FREE_ADDRESS
INT    service becomes "INT"
SET_FREE_ADDRESS address type becomes Free
COPYFMT
SRC.TS_TRUENUMBER separate address type prefix and address
":!!"
VAR.0
DST.TS_FREE_ADDR free address = address
COMPARESTRING    only succeeds if address prefix was "SMTP"
IEQU
VAR.0
"SMTP"
REM -----
REM recipient address is in FAX syntax
REM -----
ENTRY
*
SET_FREE_ADDRESS
FAX    service becomes "FAX"
SET_FREE_ADDRESS address type becomes Free
COPYFMT    separate address type prefix and address
SRC.TS_TRUENUMBER
":!!"
VAR.0
DST.TS_FREE_ADDR free address = address
COMPARESTRING    only succeed if address prefix was "FAX"
IEQU
VAR.0
"FAX"
REM -----
REM recipient address is in X.400 syntax
REM -----
ENTRY
*
SET_FREE_ADDRESS
X40    service becomes "X40"
SET_FREE_ADDRESS address type becomes Free
COPYFMT    separate address type prefix and address
SRC.TS_TRUENUMBER
":!!"
VAR.0
DST.TS_FREE_ADDR free address = address
COMPARESTRING    only succeed if address type prefix was "X400"
IEQU
VAR.0
"X400"
REM -----
REM default: (no explicit address type) assume FAX syntax
REM -----
ENTRY
*
SET_FREE_ADDRESS

```



```
SEARCH_USER_ADDRESS search KCS user with Applicom address
SRC.TS_FREE_ADDR
""
REM or REG.Setup\ServiceAC\Name
REM -----
REM Originator with KCS shadow user, via UserID
REM -----
ENTRY
AP only if service is "AP"
SET_FREE_ADDRESS only if address type is Free
*
SET_TC_ADDRESS address type becomes TC
COPYAPPEND KCS UserID = Applicom address
DST.TS_RECIP_ID
SRC.TS_FREE_ADDR
SEARCH_USER_ID search KCS user with this UserID
REM -----
REM Originator without KCS shadow user, message is sent to Fax address
REM -----
ENTRY
AP only if service is "AP"
SET_FREE_ADDRESS only if address type is Free.
*
SET_FREE_ADDRESS address type becomes Free
SET_FAX_ADDRESS only if recipient address type is Fax (additional param)
COPYAPPEND build address string:
DST.TS_FREE_ADDR $ApplicomAddress
"$"
SRC.TS_FREE_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address
REM -----
REM originator without KCS shadow user, recipient address is Telex
REM -----
ENTRY
AP only if service is "AP"
SET_FREE_ADDRESS only if address type is Free
*
SET_FREE_ADDRESS address type becomes Free
SET_TX_ADDRESS only if recipient address type is Telex
COPYAPPEND build address string:
DST.TS_FREE_ADDR $ApplicomAddress
"$"
SRC.TS_FREE_ADDR
SEARCH_LINKSVC add link queue (or service for it) to address
REM -----
REM Originator with KCS shadow user, service is EMAIL.
REM This is part of the mapfile, but not used, because APPLICOM does not
REM (yet ?) support EMAIL addresses.
REM enhanced originator mapping.
REM -----
ENTRY
EMAIL only if service is EMAIL
SET_X400_ADDRESS only if address type is X400
*
SET_TC_ADDRESS address type becomes TC
COPY copy the X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT update X400 address string
SEARCH_USER_ADDRESS search KCS user with this address
DST.TS_X400_ADDR
""
REM any service, no special service installed
REM -----
```

```

REM Originator with KCS shadow user, UserID = SurName.
REM Service is EMAIL.
REM This is part of the mapfile, but not used, because APPLICOM does not
REM (yet ?) support EMAIL addresses.
REM -----
ENTRY
EMAIL    only if service is "EMAIL"
SET_X400_ADDRESS only if address type is X400
*
SET_TC_ADDRESS address type becomes TC
COPYAPPEND UserID = SurName
DST.TS_RECIP_ID
SRC.TS_X400_SNAME
SEARCH_USER_ID search KCS user with this UserID
REM -----
REM Originator (service EMAIL) without KCS shadow user
REM This is part of the mapfile, but not used, because APPLICOM does not
REM (yet ?) support EMAIL addresses.
REM -----
ENTRY
EMAIL    only if service is "EMAIL"
SET_X400_ADDRESS only if address type is X400
*
SET_X400_ADDRESS address type becomes X400
COPY      copy the X400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT update X400 address string
SEARCH_LINKSVC add link queue (or service for it) to address

```

Section TO_TC_RECIP

- Three syntax types for source addresses: FAX, TELEX and EMAIL.
- FAX syntax: SRC.TS_SERVICE is "FX3", source address type is SET_FAX_ADDRESS, SRC.TS_FAX_NUMBER holds number and optional answerback (separated by '='). Used for Fax recipients only.
- TELEX syntax: SRC.TS_SERVICE is "TLX", source address type is SET_TX_ADDRESS, SRC.TS_TX_NUMBER holds number and optional answerback (separated by '='). Used for Telex recipients only.
- EMAIL syntax: SRC.TS_SERVICE is "EMAIL", source address type is SET_X400_ADDRESS, the X.400 address fields hold the complete address. Not yet implemented, planned to be used for all other recipient address types.

```

REM -----
TO_TC_RECIP recipient of message to KCS
REM -----
REM -----
REM recipient with Fax address
REM -----
ENTRY
FX3    only if service is "FX3"
SET_FAX_ADDRESS and if address type is Fax
*
SET_FAX_ADDRESS address type becomes Fax
COPYFMT separate number and answerback
SRC.TS_FAX_NUMBER
"=!?" separator is =
DST.TS_FAX_NUMBER
DST.TS_FAX_ANSWERB
SEARCH_MISSINGSVC use default Fax service (from config)

```

```

REM -----
REM recipient with Telex address
REM -----
ENTRY
TLX    only if service is "TLX"
SET_TX_ADDRESS  and if address type is Telex
*
SET_TX_ADDRESS  address type becomes Telex
COPYFMT  separate number and answerback
SRC.TS_TX_NUMBER
"!?"    separator is =
DST.TS_TX_NUMBER
DST.TS_TX_ANSWERB
SEARCH_MISSINGSVC use default Telex service (from config)
REM -----
REM other recipient (service EMAIL, number and service in DDA fields)
REM This syntax is not (yet ?) supported by APPLICOM
REM -----
ENTRY
EMAIL   only if service is EMAIL
SET_X400_ADDRESS and if address type is X400
*
*
SEARCH_TCADDR_IN_DDA take real address from DDA field
REM -----
REM X400 recipient (service EMAIL, KCS service in DDA field)
REM This syntax is not (yet ?) supported by APPLICOM
REM -----
ENTRY
EMAIL   only if service is EMAIL
SET_X400_ADDRESS and if address type is X400
*
*
SET_X400_ADDRESS address type becomes X400
COPY    copy the X.400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
SEARCH_TCSVC_IN_DDA take service from DDA field
UPDATE_X400TEXT update X.400 address string
REM -----
REM X400 recipient (service EMAIL, use default X.400 service)
REM This syntax is not (yet ?) supported by APPLICOM
REM -----
ENTRY
EMAIL   only if service is EMAIL
SET_X400_ADDRESS and if address type is X400
*
*
SET_X400_ADDRESS address type becomes X400
COPY    copy the X.400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
UPDATE_X400TEXT update the X.400 address string
SEARCH_MISSINGSVC use default X.400 service (from config)

```

Section TO_MAIL_ORIG

- Three syntax types for destination address: FAX, TELEX and EMAIL.
- FAX syntax: DST.TS_SERVICE must be "FX3", destination address type must be SET_FAX_ADDRESS, DST.TS_FAX_NUMBER must contain the fax address.
- TELEX syntax: DST.TS_SERVICE must be "TLX", destination address type must be SET_TX_ADDRESS, DST.TS_TX_NUMBER must contain the telex address.

- EMAIL syntax: DST.TS_SERVICE must be "EMAIL", destination address type must be SET_X400_ADDRESS, the X400 fields must contain the address.
- EMAIL syntax is not (yet ?) supported by APPLICOM.

```

REM -----
TO_MAIL_ORIG   originator of message to APPLICOM
REM -----
REM -----
REM originator address type is Fax
REM -----
ENTRY
*
SET_FAX_ADDRESS only for Fax address type
FX3   service becomes "FX3"
SET_FAX_ADDRESS address type becomes Fax
COPY  copy the address
SRC.UN_PUBLIC_ADDRESS copy the address
DST.UN_PUBLIC_ADDRESS
REM -----
REM originator address type is Telex
REM -----
ENTRY
*
SET_TX_ADDRESS  only for Telex address type
TLX   service becomes "TLX"
SET_TX_ADDRESS address type becomes Telex
COPY  copy the address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
REM -----
REM REM1 (see customization examples)
REM -----
REM -----
REM originator address type is X400
REM not (yet ?) supported by APPLICOM
REM -----
ENTRY
*
SET_X400_ADDRESS only if address type is X400
EMAIL  service becomes "EMAIL"
SET_X400_ADDRESS address type becomes X400
COPY  copy the address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
SET_ANY_DDA write service to first available DDA field type
SRC.TS_SERVICE
REM -----
REM originator address type is anything else
REM not (yet ?) supported by APPLICOM
REM -----
ENTRY
*
*
EMAIL  service becomes "EMAIL"
SET_X400_ADDRESS address type becomes X400
COPYAPPEND Surname = Shortname
DST.TS_X400_SNAME
SRC.TS_RECIP_ID
COPYAPPEND OU1 = "TCLINK"
DST.TS_X400_OU1
"TCLINK"
COPYAPPEND DDA1 type = service
DST.TS_D1NAME

```

```

SRC.TS_SERVICE
COPYAPPEND DDA1 value = address
DST.TS_D1DATA
SRC.TS_TRUENUMBER

```

Section TO_MAIL_RECIP

- Destination address uses FAX, TELEX or EMAIL syntax.
- FAX syntax: DST.TS_SERVICE must be "FX3", destination address must be SET_FAX_ADDRESS. Only used if originator is Fax.
- TELEX syntax: DST.TS_SERVICE must be "TLX", destination address must be SET_TX_ADDRESS. Only used if originator is Telex.
- EMAIL syntax: DST.TS_SERVICE must be "EMAIL", destination address must be SET_X400_ADDRESS. To be used for all other cases, but not supported by APPLICOM.
- The recipient of messages or notifications from Fax or Telex is in most cases a Free address containing the APPLICOM UserID (exception: inbound routing via rr99). Address mapping must find the Fax or Telex address for this user. Basically, it searches in the KCS recipient store.
- Recipients in messages routed via rr99 and notifications to users without KCS shadow users have a special addressing syntax ("Address"). If address mapping finds a '\$' character, it takes the rest of the string as the Fax or Telex number and makes no lookup in the recipient store.

```

REM -----
TO_MAIL_RECIP  recipient of message to APPLICOM
REM -----
REM -----
REM recipient of a message or notification from Fax
REM recipient has no KCS shadow user
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
FX3  service becomes "FX3"
SET_FAX_ADDRESS address type becomes Fax
SET_FAX_ADDRESS only if originator address type is Fax
COPYFMT  separate fullname and number
SRC.TS_TRUENUMBER (separated by '$')
"$!!"
DST.TS_FULLNAME
DST.TS_FAX_NUMBER
REM -----
REM recipient of a message or notification from Telex
REM recipient has no KCS shadow user
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
TLX  service becomes "TLX"
SET_TX_ADDRESS address type becomes Telex
SET_TX_ADDRESS only if originator address type is Telex
COPYFMT  separate fullname and number
SRC.TS_TRUENUMBER
"$!!" (separated by '$')
DST.TS_FULLNAME
DST.TS_TX_NUMBER
REM -----
REM recipient of a message or notification from Fax,
REM recipient's fax address can be found in a file.
REM This entry was only for testing purposes and is probably never

```

```

REM used in the field.
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
FX3 service becomes "FX3"
SET_FAX_ADDRESS address type becomes Fax
SET_FAX_ADDRESS only if originator address type is Fax
COPYAPPEND UserID = address
DST.TS_REC_P_ID
SRC.TS_TRUENUMBER
COPYADDRESS_LIST search file for fax address of this user
SRC.TS_TRUENUMBER UserID
F F means Fax address
.\APPLI.LST name of the file
REM -----
REM recipient of a message or notification from Telex,
REM recipient's telex address can be found in a file.
REM This entry was only for testing purpose and is probably never
REM used in the field.
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type if Free
TLX service becomes "TLX"
SET_TX_ADDRESS address type becomes Telex
SET_TX_ADDRESS only if originator address type is Telex
COPYAPPEND UserID = address
DST.TS_REC_P_ID
SRC.TS_TRUENUMBER
COPYADDRESS_LIST search file for Telex address of this user
SRC.TS_TRUENUMBER UserID
T T means Telex address
.\APPLI.LST name of the file
REM -----
REM Recipient of a message or notification from Fax.
REM Recipient has a KCS shadow user.
REM Look up his Fax address in the recipient store (for notification)
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
FX3 service becomes "FX3"
SET_FAX_ADDRESS address type becomes Fax
SET_FAX_ADDRESS only if originator address type is Fax
COPYAPPEND UserID = address
DST.TS_REC_P_ID
SRC.TS_TRUENUMBER
COPYADDRESS_STORE search user's first fax address
SRC.TS_TRUENUMBER UserID
F F means Fax
REM -----
REM Recipient of a message or notification from Telex.
REM Recipient has a KCS shadow user.
REM Look up his Telex address in the recipient store.
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
TLX service becomes "TLX"
SET_TX_ADDRESS address type becomes Telex
SET_TX_ADDRESS only if originator address type is Telex
COPYAPPEND UserID = address
DST.TS_REC_P_ID

```



```
SRC.TS_TRUENUMBER
COPYADDRESS_STORE look up user's first Telex address
SRC.TS_TRUENUMBER UserID
T   T means Telex
REM -----
REM REM2 (see customization examples)
REM -----
REM -----
REM recipient of message from Fax, recipient without KCS shadow user
REM (probably not needed any more)
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
FX3_ service becomes "FX3"
SET_FAX_ADDRESS address type becomes Fax
SET_FAX_ADDRESS only if originator address type is Fax
COPYFMT_ separate address components
SRC.TS_TRUENUMBER
"$?!"_ (separated by '$')
DST.TS_FULLNAME
DST.TS_FAX_NUMBER
REM -----
REM recipient of message from Telex, recipient without KCS shadow user
REM (probably not needed any more)
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
TLX_ service becomes "TLX"
SET_TX_ADDRESS address type becomes Telex
SET_TX_ADDRESS only if originator address type is Telex
COPYFMT_ separate address components
SRC.TS_TRUENUMBER
"$?!"_
DST.TS_FULLNAME
DST.TS_TX_NUMBER
REM -----
REM recipient has X.400 address (EMAIL syntax in APPLICOM)
REM Not (yet ?) supported by APPLICOM !
REM -----
ENTRY
*
SET_X400_ADDRESS only if address type is X.400
EMAIL_ service becomes "EMAIL"
SET_X400_ADDRESS address type becomes X.400
COPY_ just copy the X.400 address
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
REM -----
REM recipient has Free address (EMAIL syntax in APPLICOM)
REM The Free address contains an X.400 address string
REM Not (yet ?) supported by APPLICOM !
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
EMAIL_ service becomes "EMAIL"
SET_X400_ADDRESS address type becomes X.400
COPYAPPEND copy X.400 address string
DST.TS_X400_ADDR
SRC.TS_FREE_ADDR
UPDATE_X400FIELDS update X.400 address fields accordingly
REM -----
```

```

REM Recipient has Free address (EMAIL syntax in APPLICOM)
REM recipient's X400 address can be found in a file.
REM This entry was only for testing purpose and is probably never
REM used in the field.
REM Not (yet ?) supported by APPLICOM !
REM -----
ENTRY
*
SET_FREE_ADDRESS only if address type is Free
EMAIL service becomes "EMAIL"
SET_X400_ADDRESS address type becomes X.400
COPYADDRESS_LIST look up X.400 address in file
SRC.TS_FREE_ADDR UserID (the Free address)
E E means X.400 address
.\APPLI.LST file name
UPDATE_X400FIELDS update X.400 address fields accordingly

```

Customization Examples for TC/LINK-SM

This section describes customization examples for TC/LINK-SM.

Disable Address Book Lookup For Incoming Messages

When TC/LINK-SM receives a message with a local part containing not a KCS user, but a System Address Book entry, it happens that the message is sent to the fax (or whatever) number of this address book entry. If the TC/LINK-SM should only be used as a link to the Internet for KCS users, this might not be desired.

So, we design a customized MAPfile:

- Direction is "To KCS"
- We need to change the handling of active recipient addresses.

→ Therefore, we need to change the TO_TC_RECIP section! The address book lookup is done by the command "SEARCH_USER_ADDR" in the following entry:

```

ENTRY
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
ISOLATE_LOCALPART
SRC.TS_FREE_ADDR
DST.TS_RECIP_ID
SEARCH_USER_ID
REM If we remove the "ALL" here, then the Address book will not be used!
REM ALL

```

By removing the "ALL" parameter, TC/LINK-SM will no longer look for addressbook entries matching the localpart of the address. Only matching user IDs will cause the entry to succeed!

TC/LINK-SM Should Check the Complete Email Address for Distribution

If a message is addressed to an SMTPIN alias, and the same name is also an entry in the KCS system address book, then the message is sent to the address book entry, not to the SMTPIN alias.

You have two possibilities here:

- Simply disable addressbook lookup (as described in the example before)
- Change the MAP file that the SMTP recipient addresses are posted to KCS as a full address (disable the need for SMTPIN functionality; all aliases are entered as inactive SMTP addresses with full domain!).

So, we design a customized MAPfile:

- Direction is “To KCS”
- We need to change the handling of active recipient addresses.

→ Therefore, we need to change the TO_TC_RECIP section! All entries looking for two or more parameters in the localpart remain unchanged; but, before looking into the address book, we need to add a new entry:

```
ENTRY ----- search SMTP aliases with full domain first
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
SEARCH_USER_ADDRESS
SRC.TS_FREE_ADDR
""
REM Rest as usual!
ENTRY ----- Single parameter -> ...
```

Note The rule “SEARCH_USER_ADDRESS” used here requires TCROSS 7.08 or higher!

Routing to Different Fax Lines Based on Originator Address

This was an enhancement request from a customer that wants different people to use different fax channels for reasons of cost accounting. Any department has its own subdomain (e.g. <dept1.company.com>, <dept1.company.com>, ...), and its related FAX service named as the department (“dept1”, “dept2”, ... in our example).

All users shall address their mail to <number@fax.company.com>. TC/LINK-SM shall itself decide (based on the originator address), which fax service shall be used.

So, we design a customized MAPfile:

- Direction (again) is “To KCS”
- We first need to extract the subdomain of the originator address
- Store this subdomain (must be in registry as variables are non-permanent)
- Build a new recipient number by adding the subdomain as the destination service!
- As this is a pure SMTP to Fax gateway, other addressing features (like X.400, telex, other Links) are not required.

→ Therefore, we need new TO_TC_ORIG and TO_TC_RECIP sections!

```
REM *****
REM Special MAP File for Enh. #3960
REM DO NOT USE FOR OTHER SCENARIOS!
REM *****
REM -----
TO_TC_ORIG   originator of message to KCS
REM -----
```

```

ENTRY ----- Always keep originator address, extract originator dept.
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYFMT
SRC.TS_FREE_ADDR
"@???!!"
VAR.1
VAR.2
VAR.3
DST.TS_RECIP_ID
VAR.4
REM Now VAR.4 holds the domain! Extract department, store it to registry!
COPYFMT
VAR.4
".!!!"
REG.TCLSM\SubDomainStore
VAR.1
VAR.2
REM copy full address to make notifs work
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_FREE_ADDR
SEARCH_LINKSVC
REM -----
TO_TC_RECIP    recipient of message to KCS
REM -----
ENTRY ----- Localpart holds number; take service from originator!
*
SET_FREE_ADDRESS
*
SET_FAX_ADDRESS
REM Localpart holds the Fax number
ISOLATE_LOCALPART
SRC.TS_FREE_ADDR
DST.TS_FAX_NUMBER
REM Service was stored to registry previously
COPYAPPEND
DST.TS_SERVICE
REG.TCLSM\SubDomainStore
REM -----
TO_MAIL_ORIG    originator of message to Internet
REM -----
ENTRY ----- (check for SMTP address; if available, take only this!)
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
CHECKLINKQUEUE
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_TRUENUMBER
ONLYTHIS
ENTRY ----- (TC address: UserID@domain)
*
SET_TC_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_TC_USERID
"@
REG.TCLSM\SMLinkDomain

```

```

ENTRY ----- (FAX address: number@domain)
*
SET_FAX_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_FAX_NUMBER
"@
REG.TCLSM\SMLinkDomain
REM -----
TO_MAIL_RECIP    recipient of message to Internet
REM -----
ENTRY ----- SMTP recipient is always addressed as FREE address type!
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS
COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_TRUENUMBER

```

Customization Examples for TC/LINK-FI

This section describes customization examples for TC/LINK-FI.

Personal Address Book Entries

A customer requested to use the personal address book in addresses to KCS.

- Private address book shall be accessed by the TCFI syntax "TO: SN=<address book name>, SE=TOPCALL, N=<address book entry>"
- TC/LINK-FI shall first look to the private address book, then to the system address book.

So, we design a customized MAPfile:

- Direction is "To KCS"
- We need to add an entry in the TO_TC_RECIP section to look into the private address book.

```

TO_TC_RECIP -----
REM New entry for searching private address book
ENTRY ----- Look for private address book
*
SET_TC_ADDRESS
*
SET_TC_ADDRESS
REM The shortname holds the private address book name
COPYAPPEND
DST.TS_SECTION
SRC.TS_RECIP_ID
REM The number is the address book entry
COPYAPPEND
DST.TS_RECIP_ID
SRC.TS_TC_USERID
SEARCH_USER_ID
ALL
ENTRY ----- (check if KCS user or System Address)
(rest as usual)

```

Sending Copy Messages

A customer wanted to correlate pure send copy messages (without a notification) to the original recipient's correlation information.

This is possible via the combination of correct TCOSS and TCLINK version, TCOSS configuration and a map file modification.

TCOSS supports mapping of recipient correlation fields into the custom fields (TS_CORREL_1 to TS_CORREL_4) of the mail entry. This option is described in the TCOSS Application Module Manual (section 2.2.1.5 in manual version 7.64.00).

System configuration, line 20, contains four hex positions that determine the contents of the custom fields.

For this example, we assume the following configuration:

Position	Value (hex)	Comment
1	27	Recipient TS_CORREL_1
2	28	Recipient TS_CORREL_2
3	29	Recipient TS_CORREL_3
4	2A	Recipient TS_CORREL_4

With TCOSS version 7.65.00 and above, a send copy message contains the custom fields of the original message. To use these fields in address mapping, you must modify the link's map file.

This example shows how the TC/LINK-FI map file can be modified to return the original recipient's TS_CORREL_1 field in a send copy message.

Modified map file section TO_MAIL_RECIP

```

TO_MAIL_RECIP -----
ENTRY -----
*
*
*
*
COMPARESTRING
IEQU
PAR.INT_EVENT_TYPES
2056
COPY
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND
DST.TS_SERVICE
SRC.TS_SERVICE
COPYAPPEND
FORCE
DST.TS_CORREL_3
PAR.TS_CORREL_1
ENTRY -----
*
*
*
*

```

```

COPY
SRC.UN_PUBLIC_ADDRESS
DST.UN_PUBLIC_ADDRESS
COPYAPPEND
DST.TS_SERVICE
SRC.TS_SERVICE

```

The first map file entry reads the parameter PAR.INT_EVENT_TYPES (mail entry field INT_EVENT_TYPES) and checks if this is a send copy event. For every other event type, TCLINK uses the second map file entry, which is the standard behavior from previous versions.

For send copy events (event type = 2056), TCLINK reads the custom field 1 via parameter PAR.TS_CORREL_1. With the configuration described above, the content of this custom field is the original recipient's TS_CORREL_1 field. The map file rule copies this correlation field into the recipient's TS_CORREL_3 field.

Note TS_CORREL_1 and TS_CORREL_2 must not be changed because they are needed for correct termination of the send copy event.

As a result, the recipient of a send copy event has three correlation fields:

TS_CORREL_1 and TS_CORREL_2 contain the message ID of the send copy event.

TS_CORREL_3 contains the original recipient's TS_CORREL_1 information, and can thus be used to correlate the send copy to the original send order.

Customization Example for TC/LINK-AC

- In the moment, only FAX and TELEX messages can be sent to APPLICOM.
- A customer requests that they want to be able to receive messages from KCS users, too.

So, we design a customized MAPfile:

- Direction is "To Mail"
- We add an entry in the TO_MAIL_ORIG section that treats KCS users like FAX originators.
- As the recipient address depends on the originator address type, we also need an entry in the TO_MAIL_RECP section.

The following MAPfile changes treat KCS originators like FAX originators. The send service will be set to "FX3", and the originator's UserID is written to the Fax number field (in CIL).

A KCS user can then send a message to the shadow user of an APPLICOM user.

Add to section TO_MAIL_ORIG (at position marked with **REM1**)

```

REM -----
REM originator is KCS user, treated like Fax (use FX3 syntax)
REM -----
ENTRY
*
SET_TC_ADDRESS only if address type is TC
FX3 service becomes "FX3"
SET_FAX_ADDRESS address type becomes Fax
COPYAPPEND fax number = User ID
DST.TS_FAX_NUMBER

```

SRC.TS_REC_P_ID

Add to section TO_MAIL_REC_P (at position marked with **REM2**)

```
REM -----  
REM recipient has KCS shadow user, originator is KCS user  
REM (treated like incoming Fax)  
REM -----  
ENTRY  
*  
SET_FREE_ADDRESS  
FX3_ service becomes "FX3"  
SET_FAX_ADDRESS address type becomes Fax  
SET_TC_ADDRESS only if originator address type is TC  
COPYAPPEND UserID = address  
DST.TS_REC_P_ID  
SRC.TS_TRUENUMBER  
COPYADDRESS_STORE search user's first fax address  
SRC.TS_TRUENUMBER UserID  
F F means Fax
```


Restrictions / Additional Remarks

- MAPfile is in ANSI charset (you can edit it via Notepad)
- Avoid using national characters in MAPFiles.
- Maximum line length for the MAPfile is 80 characters (including CR/LF)
- Entries handling special conditions come first, entries handling defaults must be placed at the end of the section.
- All destination addresses in TO_TC_ORIG and TO_TC_RECIP must have a valid service.
- All destination addresses in TO_TC_ORIG must be either KCS shadow users (rule SEARCH_USER_ADDRESS or SEARCH_USER_ID) or routed via TCLINK (rule SEARCH_LINKSVC)
- For an explanation of the addressing syntax, take a look at the special link manual.
- Enable the TCSI trace (registry "...\\TOPCALL\\TCSIDebug" = 1) to see what comes from KCS.
- With tracelevel 0xff (registry "...\\General\\Tracelevel" = 0xff), you get a trace from the mapping process (indication of entries used, etc).
- Pay attention to the fact that there are no security checks of the registry access (by REG.xxx Entries)! Make sure to use proper arguments!
- The MAPfile is only loaded at startup of TC/LINK; therefore, you need to restart TC/LINK after any changes!
- During TC/LINK software development, new MAPFile rules and objects were added as they were needed. Therefore, the code of the older MAPFiles is sometimes more complicated than necessary.