

# Kofax Communication Server

## TC/XML Technical Manual

Version: 10.3.0

Date: 2019-12-13

The KOFAX logo is rendered in a bold, blue, sans-serif typeface. The letters are thick and closely spaced, with a consistent weight throughout the word.

© 2019 Kofax. All rights reserved.

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

# Table of Contents

<b>Chapter 1: Preface</b> .....	<b>4</b>
Further Documentation.....	4
<b>Chapter 2: TC/XML Message Format</b> .....	<b>5</b>
Message to KCS.....	6
Per-Message Sending Options.....	6
Originator.....	6
Recipients.....	6
Text Attachment.....	7
File Attachment.....	8
Custom Attachment.....	9
Linked KCS Message.....	11
Date, Time.....	11
Notification from KCS.....	12
Message from KCS.....	13
Notification to KCS.....	14
Dirsync Message.....	15
Embedded XML.....	15
<b>Chapter 3: TC XML link format (TCXL)</b> .....	<b>17</b>
Message to KCS.....	18
Per-Message Sending Options.....	18
Originator and Recipients.....	19
Text Attachment.....	21
File Attachment.....	21
Notification from KCS.....	23
Message from KCS.....	25
Notification to KCS.....	26
<b>Chapter 4: Troubleshooting</b> .....	<b>28</b>
TC/XML Message Format.....	28
TC XML link format (TCXL).....	28
Hints.....	29
Restrictions.....	29

## Chapter 1

# Preface

The purpose of this XML message format description is to explain the typical usage and the general ideas behind the XML format structures. It does not contain a complete definition of all fields with the appropriate restrictions, for this please refer to the formal schema definition.

There are two message formats supported by the XML integration:

- TC XML message format (TC/XML)
- TC XML link interface format (TCXL)

The TC XML message format (TC/XML) is an XML version of the well known TC Open Message Format, short TOM. Like TOM it allows simple messages to be represented by a simple structure, yet providing optional fields to access the full range of KCS message features if required.

The TC XML link interface format (TCXL) is an XML mapping of TC Client-Server Interface (TCSI) objects. Its structure is more complex compared to the TC/XML format. Use this format only if highest throughput performance is required.

The TCXL format is always used as an intermediate format which is easily converted into TCSI objects. The mapping between TC/XML and TCXL formats is done by applying an XSLT transformation. The XSLT transformation may be customized allowing inclusion of user-defined XML data structures in the message.

## Further Documentation

For both XML message formats a strict definition exists as

- W3C XML schema (TCXML.xsd and TCXL.xsd)
- Auto-generated word document of the schema (tcxmlschema.doc and tcxlschema.doc)

Additionally, the standard XSLT transformations between TC/XML and TCXL (TCXMLIn.xslt and TCXMLOut.xslt) may be used to clarify the relation between the two XML message formats.

## Chapter 2

# TC/XML Message Format

The TC/XML message consists of header information specifying originator, recipients and sending options, and of the message content in the form of a list of attachments.

In most cases the TC/XML format is derived from the legacy TOM format by transforming TOM sections and TOM parameters into XML elements. In some cases, for example when specifying alternative addresses, additional elements like the <ALTERNATIVE> element have to be used to get well-formed XML documents.

Unlike TOM the TC/XML format does not provide abbreviated names. Element names in XML are case sensitive, all TC/XML tags are in capital letters.

The TC/XML format uses the namespace URI "*http://www.topcall.com/XMLSchema/2002/tc/xml*". Note that the namespace URI is supposed to be globally unique. Although it follows the recommendation to use a URL based on the company's domain name, there is no implication that there is anything of interest to be found at that address. It's just an elaborate constant, not the address of something that the XML processor has to go and fetch.

Here is an example of a simple TC/XML message:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>My First Message</SUBJECT>
  <P>HIGH</P>
  <FROM>
    <SERVICE>TCFI</SERVICE>
    <NUMBER>MeOnTCFI</NUMBER>
  </FROM>
  <TO>
    <SERVICE>SMTP</SERVICE>
    <NUMBER>AI@kofax.com</NUMBER>
  </TO>
  <TXT>
high priority message
</TXT>
</MESSAGE>
```

The top-level element <MESSAGE> encloses the message, which contains header information (<SUBJECT>, <P>, <FROM>, <TO>) and message content (a single <TXT> attachment). This is not the most simple example, the <SUBJECT>, <P> (Priority) and <FROM> elements are optional and could have been omitted.

Supported XML encodings include 'UTF-8', 'UTF-16', 'ISO-8859-1', 'ISO-8859-2' and 'ISO-10646-UCS-2'.

## Message to KCS

This section describes how to send messages to KCS.

### Per-Message Sending Options

Per-message send options are specified at the beginning of the message and apply by default to all recipients. Per-recipient send options may override these settings for a specific recipient.

#### Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>Testmessage for per-message send options</SUBJECT>
  <P>HIGH</P>
  <NF>ALL</NF>
  <RESOLUTION>HIGH</RESOLUTION>
  <FROM>
    <SERVICE>TCFI</SERVICE>
    <NUMBER>TestuserOnTcfi</NUMBER>
  </FROM>
  <TO>
    <SERVICE>FAX</SERVICE>
    <NUMBER>66133899</NUMBER>
  </TO>
  <TO>
    <SERVICE>FAX</SERVICE>
    <NUMBER>004318635320</NUMBER>
  </TO>
  <TXT>
    Test for per-message send options!
  </TXT>
</MESSAGE>
```

In this example, the per-message sending options <SUBJECT>, <P> (priority), <NF> (notification request) and <RESOLUTION> (fax sending resolution) are used.

### Originator

The message originator is specified using the <FROM> element. If it is not included in the message a configurable default originator is used.

### Recipients

Recipients come in 4 varieties (to, cc, bcc and authorizer) and are accordingly specified using the elements <TO>, <CC>, <BCC> and <AUTH>. A message may have several recipients.

A single recipient consists of

- per-recipient sending options (optional)
- descriptive fields like name, department and company (optional)
- correlation information (optional)
- address

- alternative addresses (optional)

Per-recipient sending options override the per-message sending options. Descriptive fields like name, department and company may be accessed by cover variables to build a nice message cover. The correlation information is used to correlate a notification received from the KCS server to a specific recipient. Alternative addresses are activated by the KCS message server if delivery to the primary address fails.

The following example shows <TO> and <CC> recipients, different priority settings, and alternative addresses in the first recipient:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <P>NORM</P>
  <TO>
    <SERVICE>FAX</SERVICE>
    <NUMBER>66133899</NUMBER>
    <ALTERNATIVE>
      <SERVICE>FAX</SERVICE>
      <NUMBER>66133678</NUMBER>
    </ALTERNATIVE>
    <ALTERNATIVE>
      <SERVICE>SMTP</SERVICE>
      <NUMBER>AI@kofax.com</NUMBER>
    </ALTERNATIVE>
  </TO>
  <TO>
    <P>HIGH</P>
    <SERVICE>SMTP</SERVICE>
    <NUMBER>Hurry@Urgent.co.at</NUMBER>
  </TO>
  <CC>
    <P>LOW</P>
    <SERVICE>SMTP</SERVICE>
    <NUMBER>LazyBoy@sleep.com</NUMBER>
  </CC>
  <TXT>
  Sending to multiple recipients ...
</TXT>
</MESSAGE>
```

## Text Attachment

A text attachment consists of text enclosed in a <TXT> element. The text is converted from the encoding specified in the XML declaration to the configured TCOSS code page. Characters which are not found in the TCOSS code page (e.g. the tabulator control character) are replaced by a blank.

All line breaks between the start tag <TXT> and the end tag </TXT> are part of the text content. The text of the example below starts with an empty line.

The characters '<', '>' and '&' in the text (and in other string literals) are escaped using the XML built-in entities.

<	&lt;
>	&gt;
&	&amp;

Numbered character entities may be used for Unicode characters which can't be represented natively in the chosen XML encoding (or are difficult to enter because not found on the keyboard). Numbered character entities begin with the 2-character sequence `&#` followed by the ISO-10646 value of the character (decimal or hexadecimal preceded by an `x`) and end with a semicolon.

**Example:** "Fran&#231;ais"

Of course using numbered character entities makes only sense for characters, like the *c cedilla* in the example above, which are covered by the configured TCOSS code page.

## File Attachment

There are two types of file attachments, linked and embedded. Linked attachments refer to a file from which to load the binary content, while embedded attachments include the binary content in a `<BINARY>` element in Base64 encoding.

The Base64 encoding is defined in RFC 2045. Three binary bytes are represented by four bytes of a 65-character subset of US-ASCII (upper and lower-case letters, digits, `'+'`, `'/'` and `'='`). The Base64 encoded stream is arranged in lines of no more than 76 characters each.

An attachment is handled as embedded attachment if it contains a `<BINARY>` element, otherwise it is linked.

The `<APPLICATION>` element is required for both types of attachments. It contains a (dummy) file name with an extension. The extension (e.g. `'.doc'`) defines the file type and the further handling of the binary content.

The `<NAME>` element is only required for linked attachments and specifies the file from which the binary content is loaded.

The following example shows a text attachment, followed by a linked attachment and finally an embedded one (the `<BINARY>` element content may be a little longer in a real-world case):

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>TC/XML Message Format description</SUBJECT>
  <FROM>
  <SERVICE>MQ</SERVICE>
  <NUMBER>MeOnMQSeries</NUMBER>
  </FROM>
  <TO>
  <SERVICE>FAX</SERVICE>
  <NUMBER>66133831</NUMBER>
  </TO>
  <TXT>
Thanks for your interest in the KCS solution.
The complete XML link documentation follows ...
  </TXT>
  <ATT>
  <NAME>C:\TEMP\PPD_link_xml.doc</NAME>
  <APPLICATION>TOM_10600.doc</APPLICATION>
  </ATT>
  <ATT>
  <APPLICATION>format.doc</APPLICATION>
  <BINARY>sdidjeigogflg78w3hcdj3847AS7jhas==</BINARY>
  </ATT>
</MESSAGE>
```



## Custom Attachment

The custom attachment allows to include any custom XML data and format it nicely using a customized style sheet.

It is structured similar to the regular file attachment, with an <APPLICATION> element defining the type of the customized style sheet output with an extension, e.g. '.htm'.

Here is an example of a message containing a custom attachment:

```
<?xml version="1.0" encoding="UTF-8"?>
<tc:MESSAGE xmlns:tc="http://www.topcall.com/XMLSchema/2002/tc/xml" xmlns="http://
www.kcscustomer.com/XMLSchema/2002/shares">
  <tc:SUBJECT>example with custom attachment</tc:SUBJECT>
  <tc:TO>
    <tc:SERVICE>FAX</tc:SERVICE>
    <tc:NUMBER>12345</tc:NUMBER>
  </tc:TO>
  <tc:CUSTOM-ATT>
    <tc:APPLICATION>shares.htm</tc:APPLICATION>
    <tc:COMMENT>table of selected share prices</tc:COMMENT>
    <share-prices>
      <company>
        <company-name>Kofax Austria GmbH</company-name>
        <symbol>TOPC</symbol>
        <currency>EUR</currency>
        <bid>1.24</bid>
        <ask>1.40</ask>
        <high>1.40</high>
        <low>1.24</low>
        <last>1.32</last>
        <previous-close>1.30</previous-close>
        <percent-change>1.5</percent-change>
        <volume>1500</volume>
      </company>
      <company>
        <company-name>Global Graphics S.A.</company-name>
        <symbol>GLGR</symbol>
        <currency>EUR</currency>
        <bid>2.15</bid>
        <ask>2.45</ask>
        <high>2.30</high>
        <low>2.00</low>
        <last>2.15</last>
        <previous-close>1.85</previous-close>
        <percent-change>7.5</percent-change>
        <volume>2330</volume>
      </company>
    </share-prices>
  </tc:CUSTOM-ATT>
</tc:MESSAGE>
```

In this example the default name space is the one used by the custom data in element <share-prices>, the prefix 'tc' is assigned to the TC/XML message namespace.

A customized XSLT input transformation will be configured in the TC/Link-FI or -MQ which may look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

```

xmlns:tc="http://www.topcall.com/XMLSchema/2002/tc/xml"
xmlns:c="http://www.kcscustomer.com/XMLSchema/2002/shares" exclude-result-
prefixes="tc c">
<xsl:include href="TCXMLIn.xslt"/>
<xsl:template match="c:share-prices">
  <html>
    <head>
      <title>Selected Share Prices</title>
      <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
    </head>
    <body>
      <h1>Selected Share Prices</h1>
      <table border="2">
        <tbody>
          <tr>
            <th>Company</th>
            <th>Symbol</th>
            <th>Currency</th>
            <th>Bid</th>
            <th>Ask</th>
            <th>High</th>
            <th>Low</th>
            <th>Last</th>
            <th>Previous Close</th>
            <th>Change</th>
            <th>Volume</th>
          </tr>
          <xsl:for-each select="c:company">
            <xsl:sort select="c:company-name"/>
            <tr>
              <td><xsl:value-of select="c:company-name"/></td>
              <td><xsl:value-of select="c:symbol"/></td>
              <td><xsl:value-of select="c:currency"/></td>
              <td><xsl:value-of select="c:bid"/></td>
              <td><xsl:value-of select="c:ask"/></td>
              <td><xsl:value-of select="c:high"/></td>
              <td><xsl:value-of select="c:low"/></td>
              <td><xsl:value-of select="c:last"/></td>
              <td><xsl:value-of select="c:previous-close"/></td>
              <td><xsl:value-of select="c:percent-change"/> %</td>
              <td><xsl:value-of select="c:volume"/></td>
            </tr>
          </xsl:for-each>
        </tbody>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>

```

The customized XSLT transformation includes the standard transformation with the instruction

```
<xsl:include href="TCXMLIn.xslt"/>
```

and then provides a template for the custom data with

```
<xsl:template match="c:share-prices">
```

The custom data has the namespace prefix 'c' assigned in the style sheet. Note that the customer namespace URI "http://www.kcscustomer.com/XMLSchema/2002/shares" in the style sheet has to be identical to the customer namespace URI in the input document, the namespace prefix may be different.

The result of the above example is a message with share price information formatted in a table which looks somehow like this.

### Selected Share Prices

Company	Symbol	Currency	Bid	Ask	High	Low	Last	Previous Close	Change	Volume
Global Graphics S.A.	GLGR	EUR	2.15	2.45	2.30	2.00	2.15	1.85	7.5 %	2330
Kofax	TOPC	EUR	1.24	1.40	1.40	1.24	1.32	1.30	1.5 %	1500

## Linked KCS Message

A message already stored on the KCS message server may be included using the <INC> element. It contains a <NAME> element specifying the KCS message folder and name. If the folder specification is omitted the message is searched by default in the "+MAIL5V" folder.

Example of a message including a document stored in the KCS server's FIS folder:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>test with document included from FIS
    folder</SUBJECT>
  <FROM>
    <SERVICE>MQ</SERVICE>
    <NUMBER>MeOnMQSeries</NUMBER>
  </FROM>
  <TO>
    <SERVICE>FAX</SERVICE>
    <NUMBER>66133831</NUMBER>
  </TO>
  <TXT>
    Thanks for your interest in the KCS solution.
    Please see the following deployment strategy ...

  </TXT>
  <INC>
    <NAME>+MAIL5V/FDEPL27</NAME>
  </INC>
</MESSAGE>
```

## Date, Time

You can specify a send date and time with the tags <DATE> and <TIME> in the <TO> part of the recipient. See the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>Send time example</SUBJECT>
  <FROM>
    <SERVICE>TOPCALL</SERVICE>
    <NUMBER>MS</NUMBER>
  </FROM>
  <TO>
    <SERVICE>SMS</SERVICE>
    <NUMBER>06641234567</NUMBER>
    <NF>ALL</NF>
    <DATE>2007-01-02</DATE>
  </TO>
</MESSAGE>
```

```
<TIME>11:12:13</TIME>
</TO>
<TXT>
This is a test message from TC/link-XML
</TXT>
</MESSAGE>
```

The same way you can specify the tags <EXDATE> and <EXTIME> for the latest delivery date and time.

Specifying the latest delivery time relatively (like with the tag "EXTIMEREL" in TOM) is not supported for the XML formats.

## Notification from KCS

A notification for delivered, non-delivered or all messages is requested by setting the <NF> sending option to 'POS', 'NEG' or 'ALL' respectively in the message passed to KCS.

A notification has a layout similar to a regular message, the recipient of the notification is the originator of the original message, its originator is the recipient of the original message to which the notification refers.

Additionally the notification contains a <NFINFO> element holding

- message correlation information (the <MCORR> field of the original message)
- recipient correlation (the <C1>, <C2> etc. fields of the original message's recipient)
- delivery information like time of delivery and cost

The text content of the notification is built from the configured notification cover. It contains no additional information, it just presents the delivery information in a textual format. The text content may be ignored or replaced by a different text through customization of the outbound XSLT transformation.

### Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<NOTIFICATION xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>Del: TC/XML Open Message Format description</SUBJECT>
  <FROM>
    <SERVICE>FAX</SERVICE>
    <NUMBER>66133831</NUMBER>
  </FROM>
  <TO>
    <SERVICE>MQ</SERVICE>
    <NUMBER>MeOnMQSeries</NUMBER>
  </TO>
  <NFINFO>
    <STATUS>DEL</STATUS>
    <TIME>2002-03-03T16:31:00</TIME>
    <MCORR>00001639982</MCORR>
    <COST>5</COST>
  </NFINFO>
  <TXT>
TOPCALL Delivery Notification
-----
Message 00001639982 successfully sent to FAX,66133831
Time sent: 98-03-03 16:31:00
Subject: TC/XML Message Format Description
Cost: 5 for GUEST
  </TXT>
```

```
</NOTIFICATION>
```

In case of failure a non-delivery notification is returned to the originator.

It looks similar to the delivery notification shown above. However, immediate non-delivery notifications from messages to KCS that fail because of a syntax error, XML parsing error or because an attachment file is not found, may be addressed to the default originator if the originator cannot be found. In the same way the correlation parameters might be missing. There is only one notification returned regardless of the number of recipients.

## Message from KCS

The structure of a message from KCS is almost identical to that of a message to KCS.

The main differences are

- it does not contain custom attachments or linked KCS messages.
- all file attachments are either linked or embedded, depending on configuration.
- active recipients contain correlation information, which must be returned in a notification (if a notification is requested.)

### Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>Sending from Outlook Express To KCS File
  API</SUBJECT>
  <P>HIGH</P>
  <MCCORR>00001640022</MCCORR>
  <FROM>
    <SERVICE>SMTP</SERVICE>
    <NUMBER>AI@kofax.com</NUMBER>
    <ACTIVE>YES</ACTIVE>
  </FROM>
  <TO>
    <ACTIVE>YES</ACTIVE>
    <NF>NO</NF>
    <P>HIGH</P>
    <C1>00190658</C1>
    <C2>00000130</C2>
    <SERVICE>TCFI</SERVICE>
    <NUMBER>MeOnTcfi</NUMBER>
  </TO>
  <TXT>
  Hi there,
  This is a test message with attachment.
  </TXT>
  <ATT>
    <NAME>C:\TCFI\TC_TO_FI\TCFI000A.AT</NAME>
    <APPLICATION>bridge.jpg</APPLICATION>
    <COMMENT>image/jpeg</COMMENT>
  </ATT>
</MESSAGE>
```

This example contains a linked attachment. A notification to KCS to acknowledge delivery of the message is not required in this case because the <NF> element in the active recipient holds the value 'NO'. Still the

correlation elements <C1> and <C2>, which would have to be returned in a notification, are included in the recipient.

## Notification to KCS

A notification to KCS is required, if a message from KCS holds active recipients with an <NF> field set to 'ALL'. Whether notifications are requested or not is controlled by configuration (registry value TCLINKFI \Topcall\NotifMail set to 1 or 0).

The notification to KCS has a very simple structure, it does not contain an originator, recipients or any attachment.

The only child element used is the <NFINFO> element which contains

- recipient correlation elements <C1> and <C2> (required)
- delivery status element <STATUS> with 'DEL' or 'NONDEL' (required)
- additional delivery information like time of delivery and cost (optional)

A notification contains a single <NFINFO> element reporting delivery or non-delivery to a single recipient. In case that several recipients of a message require notifications, a separate notification for each recipient has to be generated.

### Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<NOTIFICATION
  xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <NFINFO>
    <C1>00040875</C1>
    <C2>00000136</C2>
    <STATUS>DEL</STATUS>
    <COST>123</COST>
    <LNOTE>delivered to User AI</LNOTE>
  </NFINFO>
</NOTIFICATION>
```

The above example notification reports delivery (<STATUS> is 'DEL') of a message to a single recipient (from which the <C1> and <C2> fields have been taken). The optional elements <COST> and <LNOTE> are also included. Take care not to pass the field length limits, e.g. the 24-character limit of the <LNOTE> element.

### Example of non-delivery notification:

```
<?xml version="1.0" encoding="UTF-8"?>
<NOTIFICATION
  xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <NFINFO>
    <C1>00AC1F40</C1>
    <C2>000001CF</C2>
    <STATUS>NONDEL</STATUS>
    <LNOTE>not delivered, no User1</LNOTE>
    <LACTION>XY</LACTION>
  </NFINFO>
</NOTIFICATION>
```

## Dirsync Message

It is possible to perform directory synchronization with TC/LINK-XML as it is possible with TC/LINK-FI and TC/LINK-MQ using the classic TOM format.

The message has to contain the tag `<DIRSYNC>1</DIRSYNC>`, and the text part of the message has to consist of the user string as in TOM format.

Example of TC/XML Dirsync message:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <DIRSYNC>1</DIRSYNC>
  <FROM>
    <SERVICE>TOPCALL</SERVICE>
    <NUMBER>any</NUMBER>
  </FROM>
  <TO>
    <SERVICE>TOPCALL</SERVICE>
    <NUMBER>any</NUMBER>
  </TO>
  <TXT>
A,Name=User5,Type=TCUSER1,COMP=DieFirma,TEL-FAX=07391234567
A,Name=User6,Type=TCUSER1,COMP=TheCompany,TEL-FAX=07391234567
  </TXT>
</MESSAGE>
```

The example shows how the users “User5” and “User6” are added to the KCS directory by using the template user “TCUSER1”.

For more information on the directory synchronization see the TC Open Message Format manual (*TOM Technical Manual*) and the general *TC/LINK Technical Manual*.

## Embedded XML

XML content that is not supposed to be interpreted by KCS can be put to the tag `<embedded-xml>`. It is on the same level as `<TXT>`. KCS creates a TC text block from the `<embedded-xml>` part and passes it on like any text content.

Especially TC/LINK-WM (MD) use this feature. An application creates Binary SMS (like ring-tones, picture messages or vcards), and puts them to an XML formatted file. TC/LINK-XML takes this message and forwards it to KCS, where it is routed to a TC/LINK-WM or TC/LINK-MD queue and sent as SMS to its final destination, a mobile device.

Please see the *Technical Manual for TC/LINK-WM and TC/LINK-MD* for details on Binary SMS Messages.

Example of a Ring-tone as TC/XML message:

```
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">
  <SUBJECT>Ring Tone</SUBJECT>
  <FROM>
    <SERVICE>TOPCALL</SERVICE>
    <NUMBER>MS</NUMBER>
```

```
</FROM>
<TO>
  <SERVICE>SMSC</SERVICE>
  <NUMBER>&lt;B1&gt;06641234567</NUMBER>
  <NF>ALL</NF>
</TO>
<EMBEDDEDXML-TXT>
<SMS>
  <CODING>BINARY</CODING>
  <UDH>06050415811581</UDH>
  <BODY>024a3a51d195cdd004001b2055059
    0610560558550548540820849900000</BODY>
</SMS>
</EMBEDDEDXML-TXT>
</MESSAGE>
```

On KCS, the message has the following text content:

```
<SMS>
  <CODING>BINARY</CODING>
  <UDH>06050415811581</UDH>
  <BODY>024a3a51d195cdd004001b2055059
    0610560558550548540820849900000</BODY>
</SMS>
```

TC/LINK-MD interprets this XML fragment as SMS and sends it as Binary SMS Message.

The <embedded-xml> tag is not restricted to content for TC/LINK-WM (MD). It can be used for all XML content that shall not be interpreted by KCS but by any application receiving this content.



## Chapter 3

# TC XML link format (TCXL)

The TC XML link interface format (TCXL) should be used only if highest throughput requirements have to be met. TCXL is processed faster than standard TC/XML because it can be directly converted to TCSI without applying an XSLT transformation.

TCXL does not support custom attachments because processing custom attachments requires the XSLT transformation step.

TCXL is a straightforward mapping of TCSI objects to XML syntax. When experimenting with TCXL it might help to use the standard XSLT transformations between TC/XML and TCXL to convert a TCXL message to/from the equivalent TC/XML representation.

A TCXL message consists of header information specifying originator, recipients and sending options, and of the message content in the form of a list of attachments.

Here is an example of a simple TCXL message:

```
<?xml version="1.0" encoding="UTF-8"?>
<set_entry_ms_mail xmlns="http://www.topcall.com/XMLSchema/2002/tcxl">
  <int_msg_type>49</int_msg_type>
  <un_content.l_env_cont>
    <set_header>
      <ts_ref>My First Message</ts_ref>
      <int_priority_to>50</int_priority_to>
      <set_entry_rs_originator.set_entry_rs>
        <l_full_addr>
          <set_full_address>
            <ts_service>TCFI</ts_service>
            <un_public_address.set_free_address>
              <ts_free_addr>MeOnTCFI##</ts_free_addr>
            </un_public_address.set_free_address>
          </set_full_address>
        </l_full_addr>
      </set_entry_rs_originator.set_entry_rs>
      <l_recipients>
        <set_entry_rs>
          <int_del_type>1</int_del_type>
          <l_full_addr>
            <set_full_address>
              <ts_service>SMTP</ts_service>
              <un_public_address.set_free_address>
                <ts_free_addr>AI@kofax.com##</ts_free_addr>
              </un_public_address.set_free_address>
            </set_full_address>
          </l_full_addr>
        </set_entry_rs>
      </l_recipients>
    </set_header>
    <obj_body_part />
    <set_att_obj>
      <int_content_type>1076</int_content_type>
```

```

<un_content.blk_binary.tctext>
high priority message
</un_content.blk_binary.tctext>
</set_att_obj>
</un_content.l_env_cont>
</set_entry_ms_mail>

```

The top-level element `<set_entry_ms_mail>` encloses the message, which contains header information (in the `<set_header>` element) and message content (a single text attachment).

This is not the most simple example, the `<ts_ref>`, `<int_priority_to>` and `<set_entry_rs_originator.set_entry_rs>` elements are optional and could have been omitted.

The `<int_msg_type>` element with value '49' and the empty `<obj_body_part>` element are required.

One can note the more complex structure of this representation compared to the equivalent example in the TC/XML message format chapter.

## Message to KCS

This section describes how to send messages to KCS.

### Per-Message Sending Options

Per-message send options are specified at the beginning of the `<set_header>` element and apply by default to all recipients. Per-recipient send options may override these settings for a specific recipient.

#### Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<set_entry_ms_mail xmlns="http://www.topcall.com/XMLSchema/2002/tcx1">
  <int_msg_type>49</int_msg_type>
  <un_content.l_env_cont>
    <set_header>
      <ts_ref>Testmessage for per-message send options</ts_ref>
      <int_priority_to>50</int_priority_to>
      <int_options>65537</int_options>
      <int_termination>786444</int_termination>
      <set_entry_rs_originator.set_entry_rs>
        <l_full_addr>
          <set_full_address>
            <ts_service>TCFI</ts_service>
            <un_public_address.set_free_address>
              <ts_free_addr>TestuserOnTcfi##</ts_free_addr>
            </un_public_address.set_free_address>
          </set_full_address>
        </l_full_addr>
      </set_entry_rs_originator.set_entry_rs>
    <l_recipients>
      <set_entry_rs>
        <int_del_type>1</int_del_type>
        <l_full_addr>
          <set_full_address>
            <ts_service>FAX</ts_service>
            <un_public_address.set_free_address>
              <ts_free_addr>004318635320##</ts_free_addr>
            </un_public_address.set_free_address>
          </set_full_address>
        </l_full_addr>
      </set_entry_rs>
    </l_recipients>
  </un_content.l_env_cont>
</set_entry_ms_mail>

```

```
    </set_full_address>
  </l_full_addr>
</set_entry_rs>
</l_recipients>
</set_header>
<obj_body_part />
<set_att_obj>
  <int_content_type>1076</int_content_type>
  <un_content.blk_binary.tctext>
Test for per-message send options!
  </un_content.blk_binary.tctext>
</set_att_obj>
</un_content.l_env_cont>
</set_entry_ms_mail>
```

In this example the per-message sending options `<ts_ref>` (subject), `<int_priority_to>` (priority for to: and authoriser recipients), `<int_options>` and `<int_termination>` are used.

The `<int_options>` and `<int_termination>` are both flag collections which have flags for specific sending options defined in the lower 16 bits. The higher 16 bits are used as mask defining which flags of the lower 16 bits are actually used. For a complete list refer to the TCXL schema.

## Originator and Recipients

Originator and Recipients use a similar structure (`<set_entry_rs>` element). Sending options and correlation information make only sense for a recipient, although it is possible to specify them also for an originator (they are ignored in this case).

The message originator is specified using the `<set_entry_rs_originator.set_entry_rs>` element. If it is not included in the message a configurable default originator is used.

An originator consists of

- descriptive fields like name, department and company (optional)
- address
- alternative addresses (optional)

A message may have several recipients.

A single recipient consists of

- delivery type (to:, cc: etc.)
- per-recipient sending options (optional)
- descriptive fields like name, department and company (optional)
- correlation information (optional)
- address
- alternative addresses (optional)

The delivery type has to be specified in element `<int_del_type>` with one of the following values:

- 1 ... to: recipient
- 2 ... cc: recipient
- 3 ... bcc: recipient

#### 4 ... authorizer recipient

Per-recipient sending options override the per-message sending options. Descriptive fields like name, department and company may be accessed by cover variables to build a nice message cover. The correlation information is used to correlate a notification received from the KCS server to a specific recipient. Alternative addresses are activated by the KCS message server if delivery to the primary address fails.

The following example shows two to: recipients (<int\_del\_type> is 1) and one cc: recipient (<int\_del\_type is 2). The first to: recipient has two alternative addresses, it uses normal priority (the default specified for all to: recipients with <int\_priority\_to> in the header). The second to: recipient has high priority because the per-recipient sending option <int\_priority>, here with value 50 = high priority, overrides the default from the header. The third recipient of type cc: is set to low priority. There is no originator in this message, so the configured default originator will be used:

```
<?xml version="1.0" encoding="UTF-8"?>
<set_entry_ms_mail xmlns="http://www.topcall.com/XMLSchema/2002/tcx1">
  <int_msg_type>49</int_msg_type>
  <un_content.1_env_cont>
    <set_header>
      <int_priority_to>49</int_priority_to>
      <l_recipients>
        <set_entry_rs>
          <int_del_type>1</int_del_type>
          <l_full_addr>
            <set_full_address>
              <ts_service>FAX</ts_service>
              <un_public_address.set_free_address>
                <ts_free_addr>66133899##</ts_free_addr>
              </un_public_address.set_free_address>
            </set_full_address>
            <set_full_address>
              <ts_service>FAX</ts_service>
              <un_public_address.set_free_address>
                <ts_free_addr>66133678##</ts_free_addr>
              </un_public_address.set_free_address>
            </set_full_address>
            <set_full_address>
              <ts_service>SMTP</ts_service>
              <un_public_address.set_free_address>
                <ts_free_addr>AI@kofax.com##</ts_free_addr>
              </un_public_address.set_free_address>
            </set_full_address>
          </l_full_addr>
        </set_entry_rs>
      </l_recipients>
      <set_entry_rs>
        <int_del_type>1</int_del_type>
        <int_priority>50</int_priority>
        <l_full_addr>
          <set_full_address>
            <ts_service>SMTP</ts_service>
            <un_public_address.set_free_address>
              <ts_free_addr>Hurry@Urgent.co.at##</ts_free_addr>
            </un_public_address.set_free_address>
          </set_full_address>
        </l_full_addr>
      </set_entry_rs>
      <set_entry_rs>
        <int_del_type>2</int_del_type>
        <int_priority>48</int_priority>
        <l_full_addr>
```

```
<set_full_address>
  <ts_service>SMTP</ts_service>
  <un_public_address.set_free_address>
    <ts_free_addr>LazyBoy@sleep.com##</ts_free_addr>
  </un_public_address.set_free_address>
</set_full_address>
</l_full_addr>
</set_entry_rs>
</l_recipients>
</set_header>
<obj_body_part />
<set_att_obj>
  <int_content_type>1076</int_content_type>
  <un_content.blk_binary.tctext>
Sending to multiple recipients ...
</un_content.blk_binary.tctext>
</set_att_obj>
</un_content.l_env_cont>
</set_entry_ms_mail>
```

## Text Attachment

A text attachment is enclosed in a <set\_att\_obj> element.

The <set\_att\_obj> element contains the following child objects:

- <int\_content\_type> element with value 1076
- <un\_content.blk\_binary.tctext> element holding the actual text content

The text is converted from the encoding specified in the XML declaration to the configured TCOSS code page. Characters which are not found in the TCOSS code page are replaced by a blank.

### Example:

```
<set_att_obj>
  <int_content_type>1076</int_content_type>
  <un_content.blk_binary.tctext>first text line
second text line ...
last text line</un_content.blk_binary.tctext>
</set_att_obj>
```

Please note that all white space, including line breaks, between the start tag

<un\_content.blk\_binary.tctext> and the end tag </un\_content.blk\_binary.tctext> is part of the text content.

## File Attachment

A file attachment is enclosed in a <set\_att\_obj> element.

The <set\_att\_obj> element contains the following child objects:

- <int\_content\_type> element with value 1024
- <ts\_appl\_id> element containing a (dummy) file name with an extension. The extension (e.g. '.doc') defines the file type and the further handling of the binary content.
- <ts\_tos\_folder> element specifying specifies the file from which the binary content is loaded. This element is only required for linked attachments.

- `<un_content.blk_binary>` element holding the actual content in Base64 encoding in case of embedded attachments. This element does not exist in linked attachments.
- `<un_content.blk_binary.utf-8>` element, used as alternative to the `<un_content.blk_binary>` element to include binary content as text

There are two types of file attachments, linked and embedded. Linked attachments refer to a file from which to load the binary content, while embedded attachments include the binary content in a `<un_content.blk_binary>` or a `<un_content.blk_binary.utf-8>` element.

An attachment is handled as embedded attachment if it contains a `<un_content.blk_binary>` or a `<un_content.blk_binary.utf-8>` element, otherwise it's linked.

Example of a linked attachment:

```
<set_att_obj>
  <int_content_type>1024</int_content_type>
  <ts_tos_folder>C:\TEMP\PPD_link_xml.doc</ts_tos_folder>
  <ts_appl_id>TOM_10600.doc</ts_appl_id>
</set_att_obj>
```

The content of an embedded attachment may be provided either in a `<un_content.blk_binary>` element using Base64 encoding or in a `<un_content.blk_binary.utf-8>` element in text form (e.g. for HTML attachments).

The content of `<un_content.blk_binary.utf-8>` elements is converted from the encoding specified in the XML declaration to UTF-8 .

Example of an embedded attachment with Base64 content (the content may be a little longer in a real-world case):

```
<set_att_obj>
  <int_content_type>1024</int_content_type>
  <ts_appl_id>format.doc</ts_appl_id>
  <un_content.blk_binary>sdiidjeigogflgdj3847AS7jhas==</un_content.blk_binary>
</set_att_obj>
```

Example of an embedded attachment with HTML content:

```
<set_att_obj>
<int_content_type>1024</int_content_type>
<ts_appl_id>shares.htm</ts_appl_id>
<ts_comment>table of selected share prices</ts_comment>
<un_content.blk_binary.utf-8><![CDATA[<html>
  <head>
    <title>Selected Share Prices</title>
  </head>
  <body>
    <h1>Selected Share Prices</h1>
    <table border="2">
      <tbody>
        <tr>
          <th>Company</th><th>Symbol</th><th>Currency</th><th>Bid</th><th>Ask</th>
          <th>High</th><th>Low</th><th>Last</th><th>Previous Close</th>
          <th>Change</th><th>Volume</th>
        </tr>
        <tr>
          <td>Algo Vision plc</td><td>AVSN</td><td>USD</td><td>0.35</td><td>0.41</td>
          <td>0.42</td><td>0.39</td><td>0.39</td><td>0.41</td>
          <td>-7.14 %</td><td>42020</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>]]>
```

```

</tbody>
</table>
</body>
</html></un_content.blk_binary.utf-8>
</set_att_obj>

```

**Note** In the above example, the HTML content is enclosed in a CDATA section so that it is handled as data and not interpreted by the XML parser.

## Notification from KCS

Whether a notification for delivered or non-delivered messages is provided by KCS is controlled by flags in the `<int_termination>` element. As described earlier the `<int_termination>` value is a flag collection which has flags for specific sending options defined in the lower 16 bits. The higher 16 bits are used as mask defining which flags of the lower 16 bits are actually used.

Notification flags in `<int_termination>`:

786432	... no notification
786436	... delivery notification
786440	... non-delivery notification
786444	... delivery and non-delivery notifications

A notification is enclosed in a `<set_entry_ms_mail>` element which has a layout similar to a regular message. The recipient of the notification is the originator of the original message.

The notification's originator is the recipient of the original message to which the notification refers. The originator also contains all originally specified recipient correlation in the `<ts_correl_1>`, `<ts_correl_2>`, ... `<ts_correl_5>` fields.

The `<int_msg_type>` element with a fixed value of '10' indicates that this is a notification.

The `<set_entry_ms_mail_orig.set_entry_ms_mail>` element holds

- message correlation information (`<ts_env_name_posted>` element)
- delivery status `<int_state>` telling whether the message was delivered or not
- additional delivery information like time of delivery and cost

The text content of the notification is built from the configured notification cover.

The example on the following page shows a non-delivery notification:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<set_entry_ms_mail xmlns="http://www.topcall.com/XMLSchema/2002/tcx1">
  <int_msg_type>10</int_msg_type>
  <ts_env_name_posted>00011280217</ts_env_name_posted>
  <ts_ref>Delivery Failure: minimum example</ts_ref>
  <set_entry_ms_mail_orig.set_entry_ms_mail>
    <int_state>2468</int_state>
    <time_action>2002-11-22T10:26:00</time_action>
  </set_entry_ms_mail_orig.set_entry_ms_mail>
  <ts_ref>minimum example</ts_ref>
  <ts_env_name_posted>00011280217</ts_env_name_posted>
  <ts_cost_center>GUEST</ts_cost_center>

```

```

<ts_duration>000000</ts_duration>
<ts_last_mda_note>no fax machine detected</ts_last_mda_note>
<ts_last_mda_action>XL</ts_last_mda_action>
</set_entry_ms_mail_orig.set_entry_ms_mail>
<un_content.l_env_cont>
  <set_header>
    <set_entry_rs_originator.set_entry_rs>
      <int_del_type>1</int_del_type>
      <ts_correl_1>recipient correlation info</ts_correl_1>
      <int_active>1</int_active>
      <int_options>16391</int_options>
      <int_termination>763</int_termination>
      <int_priority>49</int_priority>
      <ts_cost_center>GUEST</ts_cost_center>
      <l_full_addr>
        <set_full_address>
          <un_public_address.set_fax_address>
            <ts_fax_number>12345</ts_fax_number>
          </un_public_address.set_fax_address>
          <int_active>1</int_active>
          <ts_service>FAX</ts_service>
        </set_full_address>
      </l_full_addr>
    </set_entry_rs_originator.set_entry_rs>
    <ts_ref>Delivery Failure: minimum example</ts_ref>
    <l_recipients>
      <set_entry_rs>
        <int_del_type>1</int_del_type>
        <int_active>1</int_active>
        <l_full_addr>
          <set_full_address>
            <un_public_address.set_free_address>
              <ts_free_addr>UserOnFi</ts_free_addr>
            </un_public_address.set_free_address>
            <int_active>1</int_active>
            <ts_service>TCFI</ts_service>
          </set_full_address>
        </l_full_addr>
      </set_entry_rs>
    </l_recipients>
  </set_header>
  <obj_body_part/>
  <set_att_obj>
    <int_content_type>1076</int_content_type>
    <un_content.blk_binary.tctext>TOPCALL NON Delivery Notification
  </set_att_obj>
</un_content.l_env_cont>

```

```

-----
Message      : "minimum example" (ID: 00011280217)
created by   :
could NOT be sent...
to Receiver  : ()
Reason       : no fax machine detected (XL)
last Retry at : 22-NOV-2002 10:26:00
Costs        : for Costcenter: GUEST
-----

```

```

this is a test message from TC/link-XML
</un_content.blk_binary.tctext>
</set_att_obj>
</un_content.l_env_cont>
</set_entry_ms_mail>

```



## Message from KCS

The structure of a message from KCS is almost identical to that of a message to KCS.

The main differences are

- it does not contain custom attachments or linked KCS messages
- all file attachments are either linked or embedded, depending on configuration
- active recipients contain correlation information, which must be returned in a notification (if a notification is requested)
- The `<int_options>` and `<int_termination>` flag collections do not contain a bit mask in the upper 16 bits

Notification flags in `<int_termination>` value:

4 ... return delivery notification

8 ... return non-delivery notification

The following example contains a text attachment and a linked attachment.

A notification to KCS to acknowledge delivery of the message is not required in this case because the `<int_termination>` value in the active recipient does not contain the bit flags 4 and 8. Still the correlation elements `<ts_correl_1>` and `<ts_correl_2>`, which would have to be returned in a notification, are included in the recipient.

### Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<set_entry_ms_mail xmlns="http://www.topcall.com/XMLSchema/2002/tcx1">
  <int_msg_type>49</int_msg_type>
  <ts_file_name>00001640022</ts_file_name>
  <un_content.l_env_cont>
    <set_header>
      <ts_ref>Sending from Outlook Express To KCS File API</ts_ref>
      <int_priority_to>50</int_priority_to>
      <set_entry_rs_originator.set_entry_rs>
        <int_active>1</int_active>
        <l_full_addr>
          <set_full_address>
            <ts_service>SMTP</ts_service>
            <un_public_address.set_free_address>
              <ts_free_addr>AI@kofax.com##</ts_free_addr>
            </un_public_address.set_free_address>
          </set_full_address>
        </l_full_addr>
      </set_entry_rs_originator.set_entry_rs>
      <l_recipients>
        <set_entry_rs>
          <int_del_type>1</int_del_type>
          <int_active>1</int_active>
          <int_priority>50</int_priority>
          <int_termination>80</int_termination>
          <ts_correl_1>00190658</ts_correl_1>
          <ts_correl_2>00000130</ts_correl_2>
          <l_full_addr>
            <set_full_address>
              <ts_service>TCFI</ts_service>
```

```

    <un_public_address.set_free_address>
      <ts_free_addr>MeOnTcfi##</ts_free_addr>
    </un_public_address.set_free_address>
  </set_full_address>
</l_full_addr>
</set_entry_rs>
</l_recipients>
</set_header>
<obj_body_part />
<set_att_obj>
  <int_content_type>1076</int_content_type>
  <un_content.blk_binary.tctext>
Hi there,

This is a test message with attachment.
</un_content.blk_binary.tctext>
  </set_att_obj>
  <set_att_obj>
    <int_content_type>1024</int_content_type>
    <ts_tos_folder>C:\TCFI\TC_T0_FI\TCFI000A.AT</ts_tos_folder>
    <ts_appl_id>bridge.jpg</ts_appl_id>
    <ts_comment>image/jpeg</ts_comment>
  </set_att_obj>
</un_content.l_env_cont>
</set_entry_ms_mail>

```

## Notification to KCS

A notification to KCS is required, if a message from KCS holds active recipients with an `<int_termination>` field containing the bit flags 4 or 8. Whether notifications are requested or not is controlled by configuration (registry value `TCLINKFI\Topcall\NotifMail` set to 1 or 0).

The notification to KCS has a very simple structure, it does not contain an originator, recipients or any attachment. Like a message it is enclosed in a `set_entry_ms_mail` element, but with a different `<int_msg_type>` value to mark it as a notification.

The `<set_entry_ms_mail>` element contains:

- a `<int_msg_type>` element with a fixed value of '10' (required)
- a `<set_entry_ms_mail_orig.set_entry_ms_mail>` element (required)

The `<set_entry_ms_mail_orig.set_entry_ms_mail>` element holds:

- recipient correlation elements `<int_cif_id>` and `<int_cif_nr>` (required)
- delivery status element `<int_state>` with '650' or '420' (required)
- additional delivery information like time of delivery and cost (optional)

The recipient correlation elements `<int_cif_id>` and `<int_cif_nr>` are filled with the `<ts_correl_1>` and `<ts_correl_2>` values from the original message's recipient to be acknowledged, adding the prefix '0x' (to indicate that it is a hexadecimal value).

The delivery status element `<int_state>` contains '650' to indicate delivery, or '420' to indicate non-delivery of the message.

In case that delivery of a message to several recipients has to be acknowledged, separate notifications each referring to a single recipient have to be generated.

**Example:**

```
<?xml version="1.0" encoding="UTF-8"?>
<set_entry_ms_mail xmlns="http://www.topcall.com/XMLSchema/2002/tcx1">
  <int_msg_type>10</int_msg_type>
  <set_entry_ms_mail_orig.set_entry_ms_mail>
    <int_cif_id>0x00AC1F1A</int_cif_id>
    <int_cif_nr>0x000001CF</int_cif_nr>
    <int_state>650</int_state>
    <ts_last_mda_note>delivered to User AI</ts_last_mda_note>
    <ts_cost>123</ts_cost>
    <ts_duration>37</ts_duration>
  </set_entry_ms_mail_orig.set_entry_ms_mail>
</set_entry_ms_mail>
```

The above example notification reports delivery (<int\_state> is '650') of a message to a single recipient (from which the <ts\_correl\_1> and <ts\_correl\_2> fields have been taken to fill <int\_cif\_id> and <int\_cif\_nr>).

The optional elements <ts\_last\_mda\_note>, <ts\_cost> and <ts\_duration> are also included.

## Chapter 4

# Troubleshooting

This section describes how to troubleshoot errors.

## TC/XML Message Format

To verify messages or notifications in the TC/XML format:

- Check if the XML namespace declaration refers to the correct URI, e.g.  
`<MESSAGE xmlns="http://www.topcall.com/XMLSchema/2002/tc/xml">`
- Validate the input document against the schema "tcxml.xsd" using a 3rd party tool like XMLSpy
- Apply the input XSLT transform to convert the document into TCXL format using the Microsoft command line tool MSXSL (Usage: MSXSL source stylesheet [options]):  
`C:\TCOSS\TCLP>msxsl input.xml TCXMLIn.xslt -o xxx_out.xml -u 4.0`

Check the output of the XSLT transform which is in TCXL format as described in the following chapter or use it directly as input to the TC/link-XML after changing the link configuration to work without transform (registry value Options\In\XMLTransform empty)

The TC/link-XML uses a non-validating XML parser and applies the configured transformation directly to the input document. This means that the validation against the schema "tcxml.xsd" is not 100% identical to the actual processing, but it may help to find errors.

## TC XML link format (TCXL)

To verify messages or notifications in the TCXL format:

- Validate the input document against the schema "tcxl.xsd" using a 3rd party tool like XMLSpy
- Use the KCS command line tool xml2tc.exe to verify and convert the TCXL to TCSI ASCII dump format (usage: xml2tc xmlfile [tcsifile]):  
`C:\TCOSS\TCLP>xml2tc xxx_out.xml xxx.asc`

Both of the above checks do not give 100% certainty that the XML document will actually work with the TC/link-XML, but may still be useful to identify errors.

## Hints

- Possible hanger of TC/LINK-FI (XML) during start-up

After updating to TC/LP 2.07.06 it happened on our test system (Windows NT4, SP6) that TC/LINK-FI (XML) hung up during loading the style sheet. The problem did not occur when using the MSXML 4.0 SP 1 DLL's (MSXML4.DLL version 4.10.9404.0, MSXML4r.dll, MSXML4a.dll).

After updating the Internet explorer to IE6 SP1 and reinstalling the Windows Service Pack (6a) and security patches, it worked correctly, even when using MSXML 4.0 SP2 DLL's (4.20.9818.0), as they are shipped with TC/LP 2.07.08.

## Restrictions

Specifying the latest delivery time relatively (like with the tag "EXTIMEREL" in TOM) is not supported for the XML formats.