

# Kofax Communication Server

## TCLSI\_BATCH Technical Manual

Version: 10.3.0

Date: 2019-12-13

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

© 2019 Kofax. All rights reserved.

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

# Table of Contents

<b>Chapter 1: Preface</b> .....	<b>4</b>
<b>Chapter 2: Introduction</b> .....	<b>5</b>
Benefits, Strengths.....	5
System Overview.....	5
Limitations.....	6
<b>Chapter 3: Features</b> .....	<b>7</b>
Periodic Actions.....	7
Action Types.....	9
SAP Function Call (Type 0).....	11
XSLT transformation (Type 1).....	12
Write Text File (Type 2).....	12
Write Database Table (Type 3).....	12
Retries.....	12
Monitoring and Alerting.....	13
Parallel Operation.....	13
<b>Chapter 4: Installation</b> .....	<b>14</b>
Prerequisites.....	14
TC/LINK-SI.....	14
Local Computer.....	14
TCOSS Server.....	14
Setup.....	16
Instance Name.....	17
General Configuration.....	17
KCS Configuration.....	18
Windows User for Startup.....	19
Batch Configuration.....	20
<b>Chapter 5: Usage Example: Exporting SAP Master Data</b> .....	<b>21</b>
Functionality.....	21
Requirements.....	21
Configuration.....	21

## Chapter 1

# Preface

This program was written as part of the product FaxConnect for Orders, where it supports periodic export of customer and article information from SAP.

Nevertheless, it can also be used for other purposes. Once a day, TCLSI\_BATCH calls a batch of SAP functions, where the result of one call can be input for the next call. Results can be stored as XML, text file (e.g. CSV) or in an ODBC compliant database. What exactly is done depends on configuration.

The program does not access SAP directly, but uses TC/LINK-SI to call the SAP functions and report results back as a sending copy.

TCLSI\_BATCH is a console application controlled by TCSR.V.

### **Prerequisites:**

TC/LINK-SI must be installed and sending copy feature must be enabled. See *TC/LINK-SI Technical Manual* for detailed information.

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

## Chapter 2

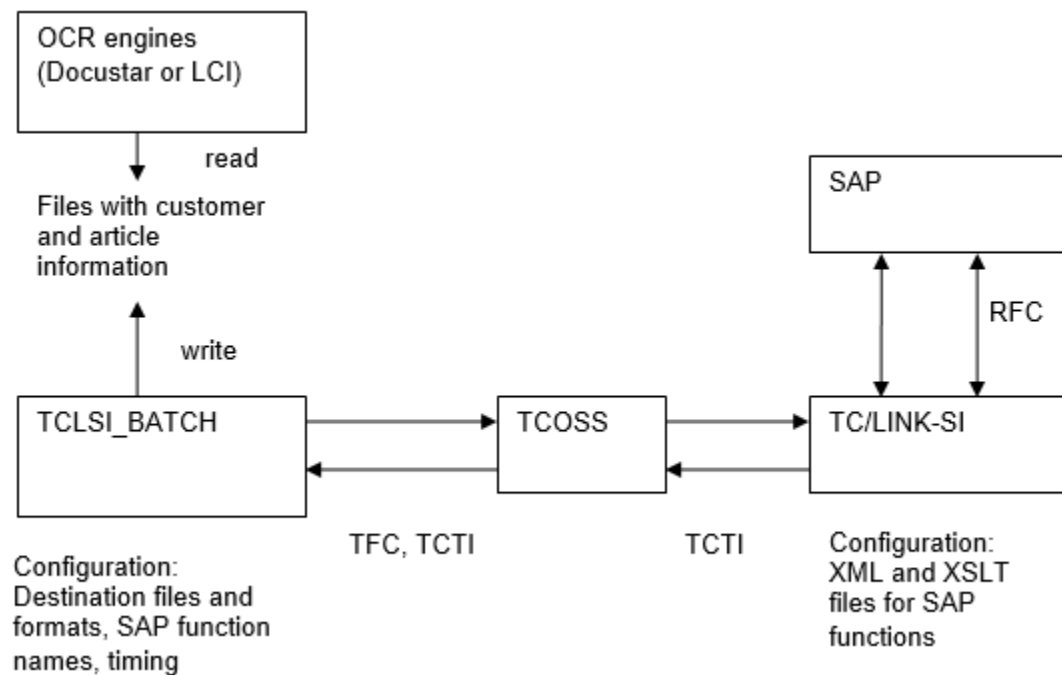
# Introduction

This section provides general information about TCLSI\_BATCH.

## Benefits, Strengths

- Serializing of SAP calls (combines several function calls done via TC/LINK-SI into one action)
- Automating of SAP calls (once a day, at a configured time)
- Highly configurable (any RFC functions can be called, any transformations possible)
- Result data can be stored as XML, CSV or in a database table
- First usage: periodic export of master data from SAP

## System Overview



## Limitations

- SAP user rights apply. TC/LINK-SI must have permission to call the SAP function and to access any data that the function needs.
- No transactions. If one call in a series of SAP calls fails, no rollback is done.

## Chapter 3

# Features

This section describes the features of TCLSI\_BATCH.

## Periodic Actions

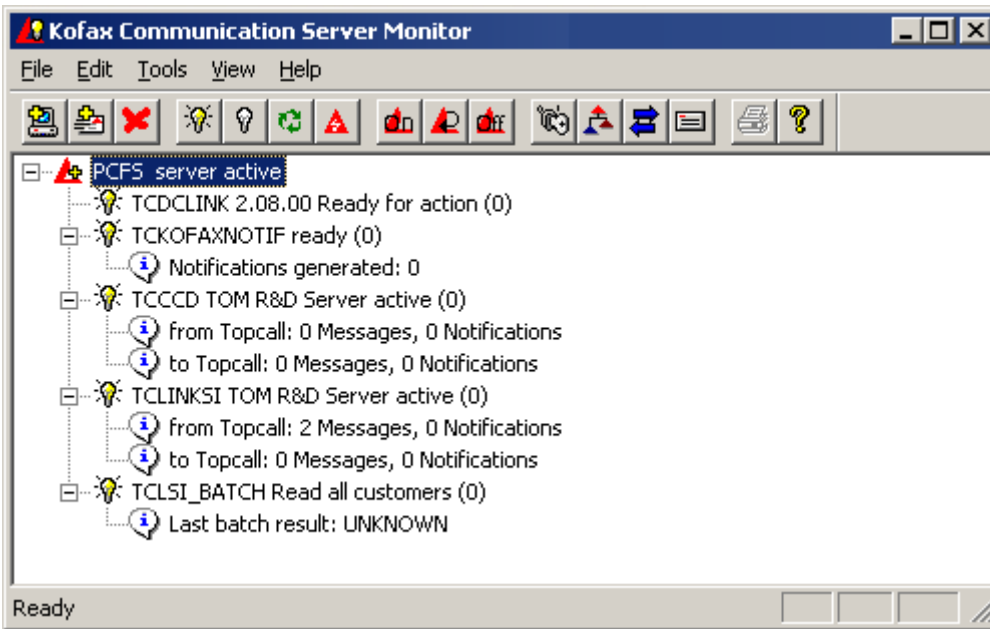
TCLSI\_BATCH runs as a sub-process of TCSR.V. It becomes active once a day, at a configured time. Then, it executes a batch of actions. The batch can consist of up to 2.147.483.647 single actions, defined in the registry or in an INI file. The single actions are numbered (Step1, Step2 etc.), the numbering reflects their sequential order.

Different types of actions (steps) are possible. The available action types are described below.

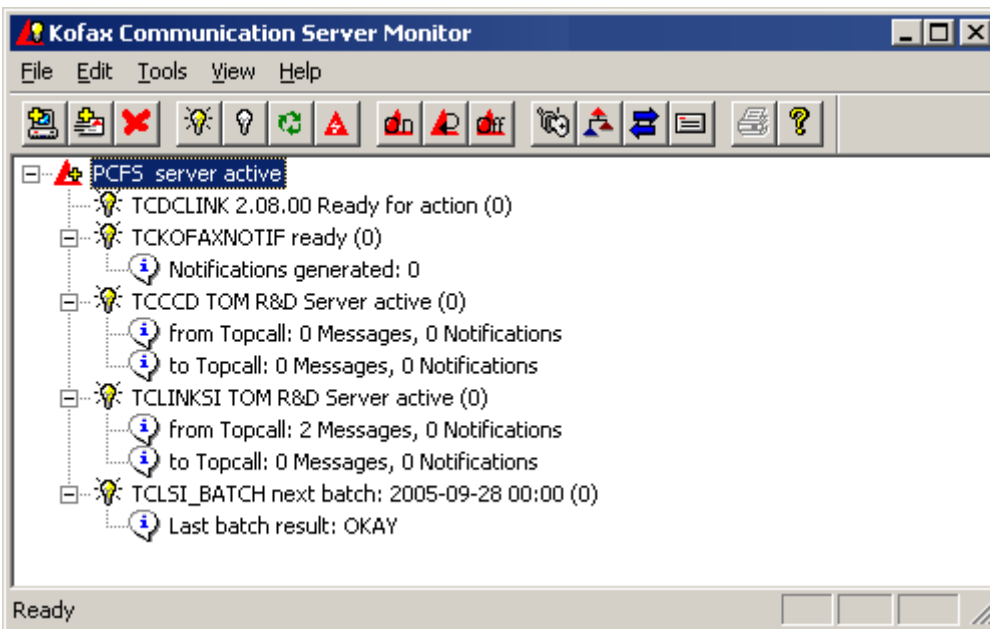
Every single step has a timeout, and will either succeed or fail. If it fails, there can be a configurable number of retries. If a step fails and no more retries are possible, the complete batch fails. Optionally, TCLSI\_BATCH will do retries of the failed batch.

Every step can yield a result, e.g. an XML file. Temporary result files are stored in a configurable folder.

Every step of the batch has a description. While the batch runs, TC/MON shows the description of the currently active step (e.g. "Read all customers"). Additionally, the overall result of the last batch is displayed (OK, ERROR or UNKNOWN right after installation).



When the program is in idle state, TC/MON shows date and time (time on PC where TCLSI\_BATCH runs) of the next batch run:



**Configuration:**

The following configuration parameters are stored in the registry below the application's subkey (HKLM \Software\Topcall\TCLSI\_BATCH).



Name	Type	Default	Description
General\BatchTime	SZ	00:00	time of day for the batch, syntax is HH:MM
General>LastBatchAt	SZ		Used internally. Time stamp of last batch execution.
General\TempFolder	SZ	C:\TCOSS\TCLP\TMP\TCLSI_BATCH	Default folder for input and result files
General\BatchDefinition	SZ		Optional: complete path name of an INI file containing the batch definition
General\PollCycle	DWORD	10	Seconds between poll attempts when waiting for a response from TC/LINK-SI
General\Retries	DWORD	0	Number of retries for failed batches.
General\WaitTimeSec	DWORD	10	Seconds between batch retries.
General\Tracelevel	DWORD	10	Trace level (set to 50 dec for a detailed trace of function calls and received messages)
Topcall\Server	SZ		KCS server name
Topcall\Path	SZ		KCS server path
Topcall\User	SZ		KCS user for logon and messaging
Topcall>Password	SZ		KCS user password (stored encrypted)
Topcall\TCCP	DWORD	0	KCS codepage
Topcall\QueueTCLINKSI	SZ		TC/LINK-SI queue

The steps of the batch (single actions) are either described in sections *Step1*, *Step2* etc of the INI file specified in registry value *BatchDefinition*, or (if this value is empty) in registry sub keys *Step1*, *Step2*, etc.

It is recommended to use an INI file, because it is much easier to change a text file than the registry. For example, see [Configuration](#).

## Action Types

Each step definition holds a Type value that can be:

Type	Description
0	SAP function call + optional XSLT transformation

Type	Description
1	XSLT transformation (XML to XML)
2	XSLT transformation (XML to text file)
3	Write XML list to database table

Additional parameters depend on the step type.

Configuration value	Used for types..	Default value	Description
InputFile	All		Optional input data (e.g. parameters for SAP function call, XML file that shall be transformed)
OutputFile	0,1,2		Output file (result of the step)
Name	All		Step description, as displayed in TCMON
Retries	All	0	Number of retries in case of error
XSLT	0,1,2		XSLT transformation to be done on the result
Function	0		Logical SAP function name
SendSubject	0		Subject of message to TC/LINK-SI (can hold function call parameters)
WaitTimeSec	0	10	Wait time between retries
TimeoutMin	0	10	Maximum wait time for notification.
XPATH	2,3		Base node for information that will be written to text file or database table
ConnectionString	3		Connection string for database
Table	3		Database table name
Append	3	0	0: existing records in table are deleted 1: existing records in table are maintained

**Note** If using an INI file for the batch definition, these parameters are below the individual step sections (Step1, Step2, etc). If the batch definition is stored in the registry, these parameters are registry values below the keys Step1, Step2 etc (*WaitTimeSec*, *TimeoutMin*, *Retries* and *Append* are REG\_DWORD, all others are REG\_SZ).

If no complete path name is specified in parameter *InputFile* or *OutputFile*, it is assumed that the file is located in the folder defined by registry key *General\TempFolder*.

## SAP Function Call (Type 0)

With this action type, TCLSI\_BATCH

- sends an SAP function call request message to TC/LINK-SI.
- waits for the notification and sendcopy from TC/LINK-SI.
- checks the notification status.
- does an XSLT transformation on the SAP function result (XML attachment of the sendcopy).
- stores the transformed result as an XML file.

### SAP Function Call Request

The following configuration values determine the contents of the SAP function call request

Registry value *Topcall\QueueTCLINKSI* holds the name of the queue polled by TC/LINK-SI (e.g. "TCLSIQI")

Parameter *Function* (defined for the step) holds the logical name of the SAP function (e.g. "BAPI\_CUSTOMER\_GETLIST")

Queue name and function name are combined to form the destination address (e.g.: "FREE,TCLSIQI:BAPI\_CUSTOMER\_GETLIST").

The subject of the function call request message is taken from step parameter *SendSubject*. It may hold a parameter for the SAP function.

Step parameter *InputFile* defines an optional file attachment for the message. The file must be in the configured folder for temporary files. Typically, this would be an XML file that resulted from a previous step.

The sender of the message is the user defined in registry value *Topcall\User*. This must be a queue user ("visible in outbox" enabled).

The message requests a delivery notification with sending copy. The sending copy will hold the function results.

A relative latest delivery timeout is set for this message (number of minutes defined in step parameter *TimeoutMin*).

### Waiting for the Notification

The program polls its queue on the KCS server, until a notification for the sent message is received. The maximum wait time is (implicitly) the configured timeout for this step.

All other messages found in the queue while waiting are ignored and terminated.

If the application is restarted during the wait time, the notification will be ignored.

## Processing the SAP Function Result

If a non-delivery notification is received, the step fails and will be either retried or will cause the complete batch to fail.

For a delivery notification, the program expects that it contains a sending copy with an XML attachment. If an XSLT transformation is defined for the step (parameter *XSLT*), the XML file is subjected to the transformation and the result is stored as a file with the (path)name defined in the parameter *OutputFile*. If no XSLT transformation is defined for the step, the XML file is just copied to the path defined in parameter *OutputFile*.

## XSLT transformation (Type 1)

With this action type, TCLSI\_BATCH does an XSL transformation (defined in parameter *XSLT*) on the *InputFile* and stores the resulting XML file at the path defined in parameter *OutputFile*.

## Write Text File (Type 2)

With this action type, TCLSI\_BATCH converts an XML file (*InputFile*) into a text file (*OutputFile*), using an XSL transformation (*XSLT*). The resulting text file may be e.g. a comma-separated or tab-separated file that can be used by another application.

It is up to the XSLT transformation script to create a text node that contains the complete resulting text. The XPATH of this text node must be configured in step parameter *XPATH*. TCLSI\_BATCH will then store the content of this node in the output file.

## Write Database Table (Type 3)

With this action type, TCLSI\_BATCH converts part of an XML file (*InputFile*) into a database table (*Table*). The database is accessed via ADODB (Microsoft ActiveX Data Objects Library, installed with Windows 2000 and above). The connection string is configured in step parameter *ConnectionString*.

Step parameter *XPATH* defines which nodes shall be stored in the database table.

**Example:** //ADDRESSDATA/ITEM

Every node becomes a record in the database table. Sub node names must match field names in the table (optional table fields need not be present in the XML file).

Typically, TCLSI\_BATCH would delete already existing records before performing this action. If step parameter *Append* is set to 1, existing records are not deleted.

## Retries

### Single step retries:

If a step fails, there can be a configured number of retries (step parameter *Retries*). The wait time between retries can be configured in step parameter *WaitTimeSec*.

Configuring retries for a step is recommended if the result file can be locked by another application, or if there are frequent network connection problems on the interface to SAP.

**Batch retries:**

If all retries for the step fail, the complete batch will be considered as failed.

TCLSI\_BATCH can be configured to retry a failed batch. Registry value *General\Retries* holds the number of retries for a failed batch (by default no batch retries are done). Before a retry, there is a wait time configured in registry value *General\WaitTimeSec*.

## Monitoring and Alerting

Errors are written into the trace file and (partially) into the application event log.

The overall status of the last batch can be seen in the TCMON status line.

With trace level (*General\TraceLevel*) 50 decimal or above, TCLSI\_BATCH writes function calls and received KCS messages into the trace file.

Currently, TCLSI\_BATCH writes the following event log messages. These events can throw SNMP traps and can therefore be used for monitoring and alerting purposes.

Code	Severity (Error, Warning, Info)	Description	Corrective Action	Parameters
1900	W	No steps defined	Configure steps in an INI file or in the registry.	
1901	W	Cannot logon to TCOSS server (%1)	Check TCOSS connection and credentials.	%1: TCSI error code
1902	I	Batch successfully completed		
1903	W	Batch failed at step %1	Check trace file for information.	%1 number of failed step
1904	I	Successfully started		
1905	I	Stopping		

## Parallel Operation

It is possible to install multiple instances of the application on the same workstation. Nevertheless, every instance must use a dedicated queue on TCOSS, and dedicated in- and output files.

Parallel operation for load sharing is not supported.

Parallel operation for failover is supported.

## Chapter 4

# Installation

This section describes the installation of TCLSI\_BATCH.

## Prerequisites

This section describes the installation prerequisites.

### TC/LINK-SI

A TC/LINK-SI instance must be installed (on any computer, connected to the KCS server that TCLSI\_BATCH uses). TC/LINK-SI must be configured to return a sending copy. See *TC/LINK-SI Technical Manual* for detailed information.

TC/LINK-SI can be installed with TC/LP 2.15.00 or higher.

### Local Computer

If the application runs with a dedicated user account, this user must be member of the local “Administrators” group and must have the right “Logon as a batch job” on the local computer.

### TCOSS Server

TCLSI\_BATCH uses a TFC license.

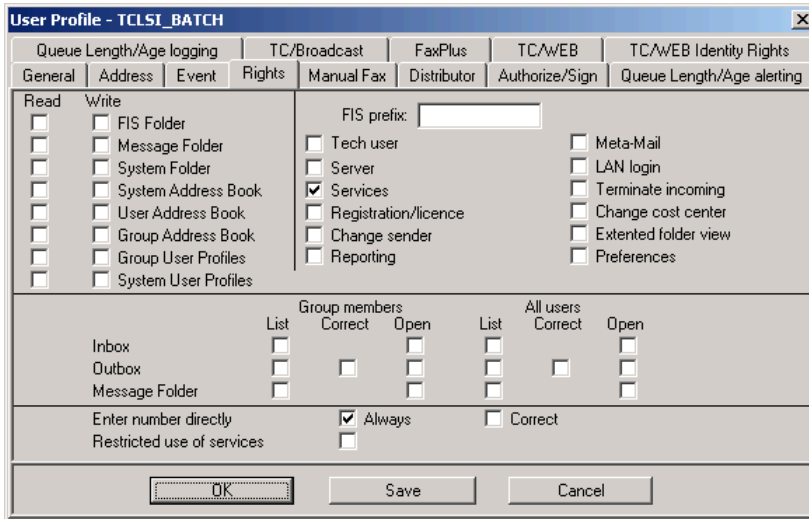
The application needs a user profile on Kofax Communication Server. This user must be created manually via TCfW.

The user profile needs the following settings:

- Visible in outbox (this is a queue user)
- It is recommended to disable dirsync and to disable password expiry for this user.
- If TC/MA is used in the system, choose an application name (e.g. TCLSI\_BATCH) for the queue and set the media type to “Topcall (for internally used queues)”.

Active	No	Service	Number
X	1	TOPCALL	TCLSI_BATCH,

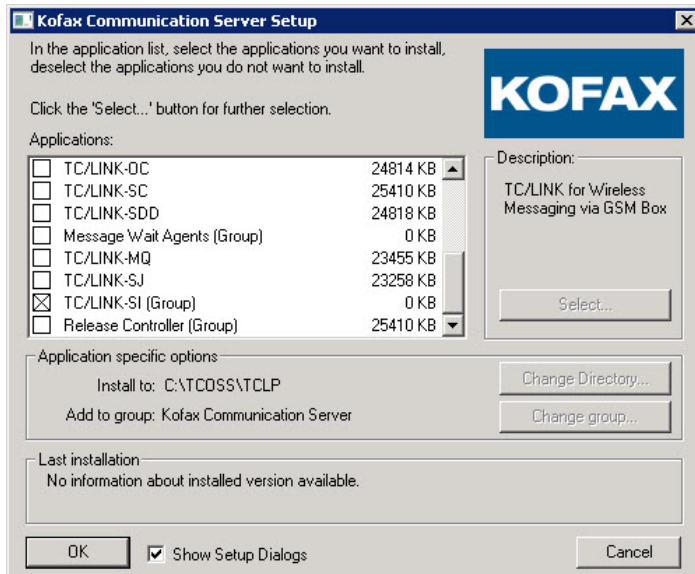
- An active address of type TOPCALL is needed.



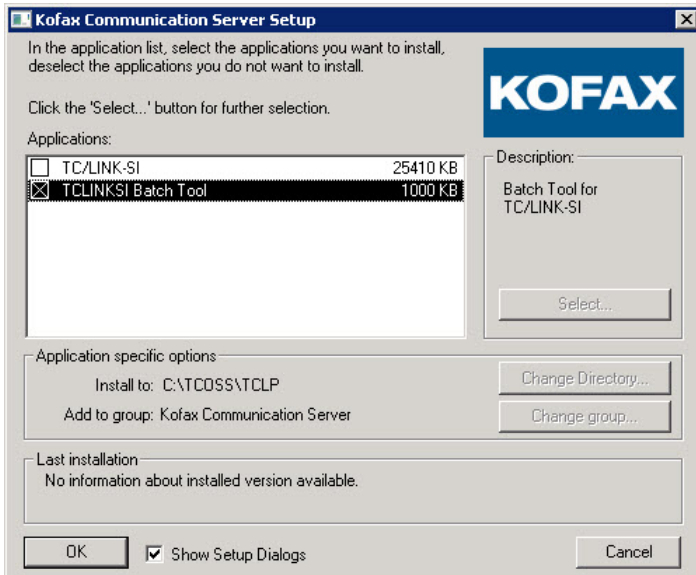
- User rights needed: Services, Enter number directly.
- The application uses only service FREE for sending. Theoretically, you could restrict the services used to just this one service.

## Setup

TCLSI\_BATCH can be installed with Kofax Communication Server Setup. In the Links group, setup contains an application group TC/LINK-SI with sub-items TC/LINK-SI and TCLSI\_BATCH.

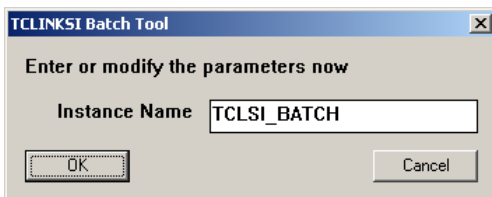






## Instance Name

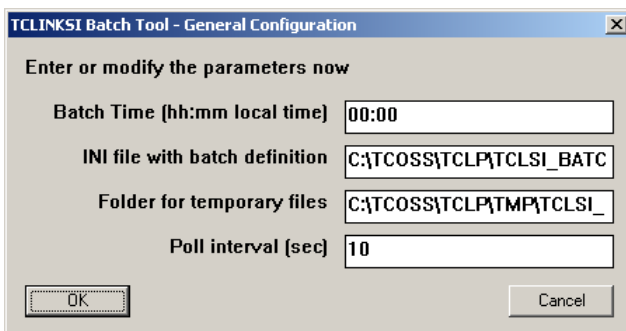
Setup asks for the name of the application (registry key name).



Setup stores the instance name in registry value *HKLM\Software\Topcall\TCLPSetup\Parameters\TCLSI\_BATCH\_Instance*.

## General Configuration

Then the most important common parameters can be configured (stored below registry subkey *HKLM\Software\Topcall\TCLSI\_BATCH\General*).



**Batch Time (hh:mm local time):** (registry: *General\BatchTime*)

Enter the time of day (based on the local workstation time zone settings) for the daily batch of actions.

Syntax: hh:mm

Default: 00:00

**INI file with batch definition:** (registry: *General\BatchDefinition*)

This input field is empty by default. Enter the complete path name of the INI file with batch definition details.

**Folder for temporary files:** (registry: *General\TempFolder*)

Enter the complete path name of the folder to be used for temporary files.

Default: C:\TCOSS\TCLP\TMP\*<InstanceName>*

**Poll Interval (sec):** (registry: *General\PollCycle*)

Enter the time interval between poll attempts when waiting for a response from TC/LINK-SI.

Default: 10

## KCS Configuration

Setup then asks for KCS connection parameters and credentials. These values are stored below registry key *HKLM\Software\Topcall\TCLSI\_BATCH\Topcall*.

The screenshot shows a dialog box titled "TCLINKSI Batch Tool - KCS Configuration". It contains the following fields and values:

KCS	TOM
KCS Path	TCP/IP,TOM
KCS User	TCLSI_BATCH
Password (default or '*' leaves existing setting)	*
TCOSS Codepage	0
TC/LINK-SI Queue	TCLSIQI

Buttons: OK, Cancel

**KCS:** (registry: *Topcall\Server*)

Enter the name of the Kofax Communication Server.

**KCS Path:** (registry: *Topcall\Path*)

Enter the path of the Kofax Communication Server.

**KCS User:** (registry: *Topcall\User*)

Enter the name of the KCS user that the application uses for logon. This must be a queue user (visible in outbox enabled).

Default: TCLSI\_BATCH

**Password:** (registry: *Topcall>Password*)

Enter the password for this user. It will be stored in an encrypted form.

Default: password

**TCOSS Codepage:** (registry: *Topcall\TCCP*)

Enter the codepage of the Kofax Communication Server.

Default: 0

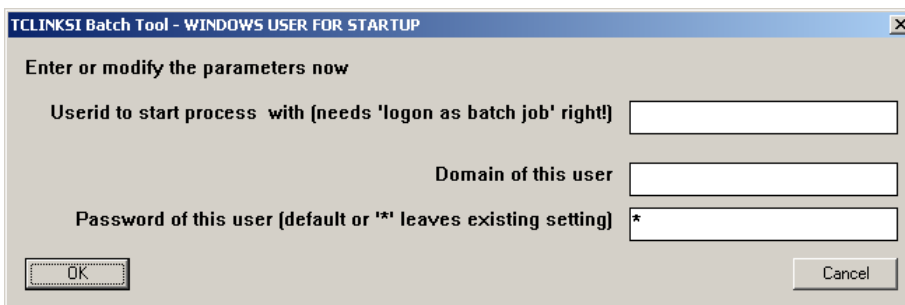
**TC/LINK-SI Queue:** (registry: *Topcall\QueueTCLINKSI*)

Enter one of the queues polled by TC/LINK-SI. The queue's image format is not important. Requests for SAP function calls are sent to this queue.

## Windows User for Startup

If TCLSI\_BATCH needs special permissions for accessing remote files or databases, you can configure a process user for the application.

To use the local system account, leave the Userid input field empty.



The screenshot shows a Windows dialog box titled "TCLINKSI Batch Tool - WINDOWS USER FOR STARTUP". The dialog contains the following text and fields:

- Header: Enter or modify the parameters now
- Field 1: Userid to start process with (needs 'logon as batch job' right!) [Empty text box]
- Field 2: Domain of this user [Empty text box]
- Field 3: Password of this user (default or '\*' leaves existing setting) [\*] [Empty text box]
- Buttons: OK, Cancel

**Userid to start process with** (registry: *UserId*):

The user must be member of the local Administrators group and must have the right to “logon as a batch job” at the local computer. Enter the name of this user.

**Domain of this user** (registry: *Domain*):

The domain of this user.

**Password of this user** (registry: *Password*):

The password of this user (stored encrypted).

## Batch Configuration

The INI file with batch definition must be maintained separately, using a text editor.

Setup copies a ZIP file with a batch example (INI file, XSLT scripts, configuration files for TC/LINK-SI) to the directory C:\TCOSS\TCLP. The file is called TCLSI\_BATCH\_SAMPLE1.ZIP.

The next section describes this sample batch scenario.

## Chapter 5

# Usage Example: Exporting SAP Master Data

This section provides an usage example.

## Functionality

The LCI project used by FaxConnectForOrders uses a customer list (in CSV format) to determine the customer IDs.

Additionally, it checks if the article numbers of the order items are valid. This is done by consulting an Access database holding article numbers and descriptions.

For line items that do not refer to existing article numbers, all values are set to '?'.

The sample batch described here provides a daily update of the customer list (CUSTOMERS.CSV) and of the MS Access database with article data (ARTICLES.MDB). The information is exported from the SAP master data.

## Requirements

TC/LINK-SI must be installed and has to run with an SAP user who is allowed to call the SAP functions BAPI\_CUSTOMER\_GETLIST and BAPI\_MATERIAL\_GETLIST.

TCLSI\_BATCH must have access to the folder where the LCI Comprend configuration is stored.

## Configuration

Extract the file TCLSI\_BATCH\_SAMPLE1.ZIP into the folder C:\TCOSS\TCLP.

The following files are copied to the local workstation.

### **Folder C:\TCOSS\TCLP\TCLSI\_BATCH\CONFIG**

#### **SAMPLE.INI**

```
[Step1]
Type=0
Function=BAPI_MATL_GETLIST
SendSubject=1000:10
OutputFile=C:\TCOSS\TCLP\TCLSI_BATCH\TEMP\MATERIAL1.XML
```

```

XSLT=C:\TCOSS\TCLP\TCLSI_BATCH\CONFIG\MATERIALS1.XSLT
InputFile=
TimeoutMin=10
Name=Read Articles for SalesArea and DistributionChannel
[Step2]
Type=3
Name=Build table of articles
InputFile= C:\TCOSS\TCLP\TCLSI_BATCH\TEMP\MATERIAL1.XML
ConnectionString=PROVIDER=MICROSOFT.JET.OLEDB.4.0;DATA SOURCE=C:\TCOSS\TCLP\LCI
\ARTICLES.MDB
Table=ARTICLES
Append=0
Retries=1
WaitTimeSec=60
XPath=//MATNRLIST/ITEM
[Step3]
Type=0
InputFile=
OutputFile=C:\TCOSS\TCLP\TCLSI_BATCH\TEMP\CUSTOMERS.XML
XSLT=
Function=BAPI_CUSTOMER_GETLIST
TimeoutMin=10
Name=Read all customers
[Step4]
Type=2
Name=Build Customers CSV file
InputFile= C:\TCOSS\TCLP\TCLSI_BATCH\TEMP\CUSTOMERS.XML
XSLT=C:\TCOSS\TCLP\TCLSI_BATCH\CONFIG\CUSTOMERS1.XSLT
OutputFile=C:\TCOSS\TCLP\LCI\CUSTOMERS.CSV
XPath=//CUSTOMERS
Retries=1
WaitTimeSec=60
    
```

The file names printed in bold letters refer to the result files used by LCI Comprend. They must of course match the LCI Comprend project configuration. If you change their names or location, you must do this in all places shown in the following table.

File	TCLSI_Batch configuration	LCI project configuration
ARTICLES.MDB	SAMPLE.INI, Step2	Script functions called IsValidItem
CUSTOMERS.CSV	SAMPLE.INI, Step4	Project settings, Databases, Definition of database Customers

### MATERIALS1.XSLT

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
...<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
...<xsl:template match="/">
  <MATNRLIST>
    ...<xsl:apply-templates select="//MATNRLIST/ITEM"/>
  </MATNRLIST>
</xsl:template>
<xsl:template match="ITEM">
  <ITEM>
    ...<xsl:copy-of select="MATERIAL"/>
    ...<xsl:copy-of select="MATL_DESC"/>
  </ITEM>
</xsl:template>
</xsl:stylesheet>
    
```

### CUSTOMERS1.XSLT

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
...<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
...<xsl:template match="/">
.....<CUSTOMERS>
    <xsl:text>&#13;&#10;</xsl:text>

    <xsl:text>&#13;&#10;CUSTOMER;SORT1;NAME;COUNTRY;COUNTRYISO;REGION;POSTL_COD1;CITY;STREET;ADDRESS;
xsl:text>
    <xsl:text>&#13;&#10;</xsl:text>
    <xsl:apply-templates select="//ADDRESSDATA/ITEM"/>
    <xsl:text>&#13;&#10;</xsl:text>
.....</CUSTOMERS>
...</xsl:template>
<xsl:template match="ITEM">
<xsl:text></xsl:text><xsl:value-of select="CUSTOMER"/><xsl:text></xsl:text><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (SORT1)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (NAME)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (COUNTRY)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (COUNTRYISO)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (REGION)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (POSTL_COD1)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (CITY)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (STREET)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (ADDRESS)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (TEL1_NUMBR)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:value-of select="normalize-space (FAX_NUMBER)"/><xsl:text></xsl:text></xsl:text></xsl:text><xsl:text>&#13;&#10;</xsl:text>
</xsl:template>
</xsl:stylesheet>

```

**Folder C:\TCOSSITCLP\TCLSI\_BATCH\TEMP**

Empty folder, will hold temporary files.

**Folder C:\TCOSSITCLP\LCI****ARTICLES.MDB**

This MS Access database file holds the ARTICLES table, with the following fields:

- MATERIAL (Text 50)
- MATL\_DESC (Text 50)

The file CUSTOMERS.CSV will be created in this folder.

**Folder C:\TCOSSITCLP\BAPI**

The following configuration files for SAP functions are installed:

- BAPI\_MATL\_GETLIST\_DEF.XML

- BAPI\_MATL\_GETLIST\_REQUEST.XSLT
- BAPI\_MATL\_GETLIST\_RESPONSE.XSLT
- BAPI\_CUSTOMER\_GETLIST\_DEF.XML
- BAPI\_CUSTOMER\_GETLIST\_REQUEST.XSLT
- BAPI\_CUSTOMER\_GETLIST\_RESPONSE.XSLT

#### **TCLSI\_BATCH Registry**

Set registry value General\BatchDefinition to C:\TCOSS\TCLP\TCLSI\_BATCH\CONFIG\SAMPLE.INI.