Kofax Communication Server

TCTI Configuration Manual

Version: 10.3.0

Date: 2019-12-13

KOFAX

Legal Notice

Table of Contents

Chapter 1

Preface

This section provides an overview of TCTI.

What TCTI Does

TCTI is a software module that takes care of the network communication between Kofax Communication Server client applications (such as TCfW) and server (such as TCOSS).

TCTI does not communicate directly with the network hardware but uses the standard network communication functionality of network clients for this. Examples of network clients are the Novel Netware Client (used with a Novel Netware Server), the IBM LAN Requester (used with an IBM LAN Manager Server), Windows for Workgroups (used with Windows Server or compatible server), or an NFS Client (used for most UNIX servers).

Network clients communicate with a server through a protocol. Most commonly used protocols are **IPX/SPX**, **NETBIOS** and **TCP/IP**. A server can support more than one protocol. For example, a Novel Netware Server supports IPX/SPX and TCP/IP. A Windows Server supports all three protocols. Both network client and network server need to install the same **protocol drivers** in order to communicate with each other. The protocol driver then communicates with a **LAN adapter driver** which does the actual communication with the LAN.

The **protocol drivers** are not only used by the network client but can also be used by other applications that do client/server communication. TCTI uses these drivers to allow Kofax Communication Server clients to communicate with servers.

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.

How TCTI Operates

TCTI optimizes the information that is transmitted between client and server for transportation across a network. It uses the Windows Remote Procedure Call (RPC) layer in order to support different network protocols like TCP/IP, IPX/SPX and NETBIOS. Furthermore, connections to local servers are supported by protocol type LOCAL.

Requirements for Client Usage

The client side of TCTI is supported for the same operating systems as the KCS Client Applications and KCS links. See the *KCS Client Applications Installation Manual* and the *TC/LINK Technical Manual* for supported operating systems.

Requirements for Server Usage

For server operation TCTI requires one of the operating systems supported for the Kofax Communication Server. See the *Platform System Manual* for details on supported operating systems for the KCS core components.

All three protocols (TCP/IP, NETBIOS, IPX/SPX) are supported with the TCTI RPC transport.

Chapter 2

Configuration

This section provides configuration information.

What Should Be Configured

The amount of configuration required for TCTI to work properly depends on whether TCTI is used by a Client or a Server application.

On the Server the only thing that must be configured is which protocols are actually used. Optionally, more advanced parameters can be set to control how the single protocols behave.

The RPC transport must receive a **connect path** from the Client application. This connect path is required by TCTI to find the **server application** on the network. It contains information about the protocol that is used to communicate, the network address of the server and the type of server application identified by the **endpoint**. Normally, the Client application provides only a network address to TCTI. TCTI then uses defaults for the missing information.

RPC and Native Transport

Native mode was the earlier implementation. It is recommended to use RPC transport now.

Note For Kofax Communication Server 10.2 and later, TCTI does not support native transport mode.

Location of the Configuration

For clients, the configuration parameters of TCTI are located in the registry key of the process which uses tcti32.dll in the subkey TCTI. For instance:

```
HKEY_LOCAL_MACHINE\Software\TOPCALL\TCLT\TCTI
```

If a parameter cannot be found there it is read from the default location which is:

```
HKEY_LOCAL_MACHINE\Software\TOPCALL\Common\TCTI
```

For the TCTI server (TCOSS or TC/Archive) the configuration parameters are located in the registry key of the process that uses the tcti32.dll. For instance:

```
HKEY_LOCAL_MACHINE\Software\TCOSS\tctiServer\rpc
```

Basic Configuration

The next topics give an overview of the parameters that can or, in some cases, must be provided for a standard configuration.

Transport Selection

With Kofax Communication Server 10.2 and later, this value is ignored because native mode is no longer used.

Name	Type	Data
Transport	REG_SZ	RPC

Protocol Selection

RPC uses LinkTypes to find the default protocol in case no protocol was specified as part of the **connect path** (provided by the application).

Name	Type	Data
LinkTypes	REG_SZ	IPX/SPX,NETBIOS,TCP/IP

For Servers, this would enable all protocols. Note that the default value is IPX/SPX.

Connect Path

The connect path is normally specified by the Client application. For TCfW Communication Server Client, the connect path to the Kofax Communication Server server is specified in HKLM\Software\TOPCALL\COMMON\Server1 for the first TCOSS server. Key HKLM\Software\TOPCALL\COMMON\ArcServer1 contains the connect path to the optional TC/ARCHIVE server that is configured for the first TCOSS server.

Example for TCfW:

Full path of Name	Type	Data
HKCU\Software\TOPCALL\COMMON\Server1\Path	REG_SZ	KSC
HKCU\Software\TOPCALL\COMMON\ArcServer1\Path	REG_SZ	KSC:ARCHIVE
HKCU\Software\TOPCALL\COMMON\TCTI\LinkTypes	REG_SZ	TCP/IP

The syntax for the connect path is as follows:

```
protocol,server address:endpoint
```

Protocol is either **IPX/SPX**, **NETBIOS** or **TCP/IP**. The server address is protocol dependent. For IPX/SPX and NETBIOS, it is the server name (such as **KCS** or **MAILSERVER**). For TCP/IP, it is either the server's IP address (such as **207.68.137.65**) or it is a DNS name (such as **kcs.mydomain.com**). The endpoint alias name **ARCHIVE** must be specified if the connect path refers to a TC/ARCHIVE server. See Advanced Configuration for more information on endpoints.

The protocol name (including terminating comma) and endpoint (including preceding colon) are optional. A protocol only needs to be specified if a different one then the TCTI default is used. An endpoint must be specified if either the server is not a TCOSS server or when the default endpoint used by TCTI collides with endpoints used by other non-KCS applications. See *Advanced Configuration* for more about endpoints.

In installations where a failover server is available (KCS Model x65), the connect path of the failover server must be specified in the connect path separated by a vertical bar '|' (the *or* sign).

Single KCS:	**DEMOTC**
`Model x65:`	`NETBIOS,KCS1|NETBIOS,KCS2`
`Archive server:`	`DEMOARC:ARCHIVE`
`non standard endpoint:`	`TCP/IP,demotc.kofax.com:251`

Advanced Configuration

This section provides information about advanced configuration.

Local Protocol

It is possible to use a special "protocol" named **LOCAL**. This LOCAL protocol does not use the network drivers but uses a special form of RPC communication called "Local Remote Procedure Call. "

The LOCAL protocol must be specified in the list of enabled protocols in the TCTI server configuration and must be specified as protocol in the connect path of the client. The client does not need to specify a network address but can optionally specify an endpoint.

Examples of valid connect paths are listed here.

Path	**Used for connection to**
LOCAL,	TCOSS server
LOCAL,:ARCHIVE	TC/ARCHIVE server

Endpoints, Ports and Sockets

Endpoints are used to identify the type of server application. Each server application must have a different endpoint in order to allow multiple server applications on one server. Depending on the protocol, endpoints are also known as ports (TCP/IP) or sockets (IPX/SPX).

TCTI uses alias names for endpoints to free the user of the burden of remembering the actual endpoint used to identify a specific server application. Endpoints are actually protocol dependent, which requires different endpoints for different protocols. For TCP/IP and IPX/SPX, the endpoint consists of a 16-bit number. With NETBIOS, the endpoint is an 8-bit number. The LOCAL protocol uses a name to specify the endpoint.

Currently, there are two alias names defined within TCTI: TCOSS and ARCHIVE. The following table shows the actual endpoints used with these alias names.

	TCOSS	ARCHIVE
IPX/SPX	64508	64509
TCP/IP	64508	64509
NETBIOS	252	253
LOCAL	TCTI	TCTIARC

A **TCTI client** selects the required server in the connect path (see above). It can either specify one of the alias names or a custom endpoint. A custom endpoint is always protocol depended.

The following example shows a custom endpoint for a TCfW client:

Name	Type	Data
Server1\Path	REG_SZ	TCP/IP,TC.MICROSOFT.COM:12345

In the case of an IPv6 address of the server:

Name	Type	Data
Server1\Path	REG_SZ	TCP/IP,[xx:xx:xx:xx]:12345

A **TCTI server** must specify its server type in the server's TCTI configuration, unless it uses the default server type "TCOSS". Here is an example for TC/ARCHIVE:

Name	Type	Data
ServerType	REG_SZ	ARCHIVE

A server that requires custom endpoints can specify a different endpoint for each protocol:

Name	Type	Data
TCPPort	REG_SZ	12345
SPXService	REG_SZ	12345
NBEndpoint	REG_SZ	123
LRPCEndpoint	REG_SZ	MYSERVER

The preceding example shows endpoint overrides for TCP/IP (TCPPort), IPX/SPX (SPXService), NETBIOS (NBEndpoint) and LOCAL (LPRCEndpoint).

Note To support different server applications on one server (such as TCOSS and TC/ARCHIVE), each server application must use a different TCTI configuration. See the documentation of the server application for more information on specifying the location of the TCTI configuration for that application.

Note In case of an IPV6 address, the endpoints should use the other from for setting of port:

[<IPV6 address>**]**:<port>

Encrypted communication

Encryption of TCTI traffic is based on Windows data privacy for RPC. Additionally, the client identity must be authenticated by the server machine, so that the encryption is supported only for the following five cases:

1. The current client process identity is a local user that exists as a local user with the same password on the server machine.
2. The current client process identity is a Windows domain (Active Directory) user and the server trusts this domain.
3. The client has configured the user ID and password of a local user on the server machine.
4. The client has configured the domain, user ID and password of a Windows domain (Active Directory) user and the server trusts this domain.
5. The client uses an anonymous logon. The option can be used only if the client operating system is Windows 8/2012 or higher.

Additional notes for cases 3 and 4: The configured user does not need to have a user profile or any permission on the client machine, **but take care that this user has very restricted permissions because the password can be disclosed**. Such a user is also required if a client process runs as a service user account (Local Service, Network Service or Local System.) Minimum permissions for this user:

- File permissions:
  - No file permissions are required.
- Security Settings -> Local Policies -> User Rights Assignment:
  - Allow “Access this computer from the network” (this is enabled by default via everyone group)
  - Verify that this user not part of “Deny access to this computer from the network”

Configuration parameters

With the default configuration, if the current client user is known by the server machine (case 1 and case 2), the entire TCSI traffic will be encrypted. Else, a fallback to none-encrypted mode is used.

- If RpcClientUserMode is set to 1, the user configured in RpcClientUserId and RpcClientUserPwd must be known by the server in order to activate encryption (case 3 and case 4.)
- If RpcClientEnableEncryption is set to 0, encryption is never used. Note that this mode fails, if the server forces encryption.
- If RpcClientEnableEncryption is set to 2, encryption is forced by the client. This mode will fail if authentication fails.
- If RpcServerForceEncryption is set to 1, encryption is forced by the server. This mode will fail if authentication fails.

Values used by clients

The detailed behavior for clients (for example, TCfW, TC/LINK-xx …) can be defined by the following TCTI registry values:

[REG_DWORD] RpcClientEnableEncryption, default: 1

0 …. Do not use encryption

1 …. Try encryption but allow fallback to none-encrypted mode (default)
The client starts with encrypted mode. If any connection attempt fails with an access denied error, the mode is changed to from encrypted to unencrypted (or vice versa) followed by a single retry.

2 …. Force encryption

[REG_DWORD] RpcClientUserMode, default: 0

0 …. Use identity of current client process (for example, interactive user with TCfW, default)

1 …. Use fixed user credentials as configured in RpcClientUserId and RpcClientUserPwd. This user is used for RPC encryption only. It is never used for LAN login.

[REG_STRING] RpcClientUserId, default: “Kofax-KCS-Rpc-User”

Optional client user for authentication if RpcClientUserMode is set to 1. On Windows 8/2012 or higher, you can also use an empty content, which means anonymous logon. It’s neither recommended nor supported to use anonymous logon with Windows 7/2008 or older!

Format: Domain\User or User.

[REG_STRING] RpcClientUserPwd

Password for fixed client user if RpcClientUserMode is set to 1. If this values does not exist, a hardcoded password (“3-4Girk!9DD_f;”) is used. If a password is entered it will be encrypted during next access. It is never used and encrypted if RpcClientEnableEncryption or RpcClientUserMode are zero.

Values used by server applications

The detailed behavior for servers (TCOSS or server part of TC/Archive) can be defined by the following TCTI registry values:

[REG_DWORD] RpcServerForceEncryption, default: 0

0 …. Accept connections from clients with and without encryption

1 …. Accept connection from clients with encryption but deny connections from client without encryption.

Configuration examples

Typical configuration values are shown in the following table.

Configuration on client	Configuration on server	Description
<default>	<default>	Encryption is used if the client machine is authenticated by the server.
RpcClientUserMode=1 RpcClientUserId=””	<default>	Encryption is used if the client operating system is Windows 8/2012 or higher.
RpcClientUserMode=1	Create Windows user “Kofax-KCS-Rpc-User” with password “3-4Girk!9DD_f;”.	Encryption is used. Client authenticates with the default user. It can be used with all supported Windows versions but it may add a security risk due to the default Windows user/password.

Configuration on client	Configuration on server	Description
RpcClientUserMode=1 RpcClientUserId=x RpcClientUserPwd=y	Create Windows user x with password y	Encryption is used. Client authenticates with a non-default user. It can be used with all supported Windows versions.

Note <default> means that registry values either do not exist, or they have their default content.

Encrypted Communication with KIC Message Connect

TCTI encryption is also supported on the network connection between Kofax Import Connector Message Connector and Kofax Communication Server (as fax server).

Encryption is used by default, because Kofax Import Connector 2.6.0.0 if the client machine is authenticated by the server.

The following registry key may be used for advanced client configurations (such as RpcClientUserMode=1, RpcClientUserId="") as described above.

- HKLM\Software\[Wow6432Node\]TOPCALL\MC\TCTI for default Message Connector instance
- HKLM\Software\[Wow6432Node\]TOPCALL\MCxx\TCTI for Message Connector instance xx

This registry key may also be used for other TCTI settings such as TraceLevel.

Compatibility

TCTI encryption is supported since following product / module versions:

- Kofax Communication Server 10.1
- Kofax Import Connector 2.6
- Kofax Import Connector Message Connector 3.26.00
- TCTI32.dll 2.16.10

The following compatibility hints must be considered in a mixed environment where both applications with (new) and without (old) support of encryption are used.

- Old clients are compatible with a new servers, if the RpcServerForceEncryption is not set to 1 on the servers.
- New clients are compatible with an old server, if the value RpcClientEnableEncryption is not set to 2 on the clients.

Support of Multiple Network Adapters

No special TCTI configuration is required for support of multiple adapters with RPC transport.

If two networks are connected via a router capable of routing the configured protocol, clients may be configured with alternative KCS paths.

Obsolete Configuration Values

The following values were used for “native” mode, which is no longer supported. They are silently ignored:

- Transport (RPC mode is always used)
- OwnNetbiosName (not required for RPC mode)
- Mode (not required for RPC mode)
- AdapterNumber (not required for RPC mode)

Chapter 3

Troubleshooting

This section provides information to troubleshoot TCTI.

NETBIOS Problems

The most common problem with the NETBIOS protocol is that the LANA number (see above) is wrong. The result is that the client cannot connect to the server. See **NETBIOS** and **LANA numbers** for information on where this is configured.

TCP/IP and Firewall

If the client and the server are separated by a firewall (router with packet filtering), it may be required to remove the TCP/IP ports used by TCTI from the firewall filter. See Endpoints, Ports and Sockets for the list of default TCP ports used by TCTI.

Trace Facility

In most cases, a TCTI trace is required to find the actual cause of the problem. The Trace can be found in the trace file of the application using TCTI (TCfW).

Trace Level

The TCTI trace for the client side can be activated by setting the following registry value:

```
HKCU\Software\TOPCALL\Common\TCTI\TraceLevel
```

For the server side (TCOSS) TCTI trace (RPC transport), set the following value:

```
HKLM\Software\TOPCALL\TCOSS\tctiServer\rpc\TraceLevel
```

The type of value should be REG_DWORD (to enter hexadecimal), but for compatibility reasons, REG_SZ (to enter decimal) is also possible.

TraceLevel specifies a 16-bit value where each bit activates a separate trace level. The definition of the trace levels is dependent on the transport type.

Level	Hex	RPC transport
1	0x1	general information or error trace

Level	Hex	RPC transport
2	0x2	connect/disconnect
4	0x4	Frames
8	0x8	API calls
16	0x10	Events
32	0x20	RPC calls
64	0x40	Reserved
128	0x80	additional information
256	0x100	Reserved
512	0x200	Reserved
1024	0x400	Reserved

Normally, in case of problems, all trace levels should be enabled by specifying **TraceLevel 2047 (0x7ff)**.