

Kofax Communication Server

Voice Platform Technical Manual

Version: 10.2.0



© 2018 Kofax. All rights reserved.

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

Table of Contents

1. HOW TO READ THIS DOCUMENT	9
2. INTRODUCTION.....	10
3. FUNDAMENTAL DEFINITIONS	11
4. CONCEPTS AND ARCHITECTURES.....	12
4.1 OVERVIEW	12
4.2 THE CONCEPT OF PLUG-INS	14
4.2.1 <i>Active Vs Passive Plug-in..</i>	15
4.3 ALIASES	15
4.4 TRS – TRANSPARENT ROUTING SCRIPT.....	16
4.5 LSR – LOCAL SESSION REDIRECTION	16
4.6 CALL TRANSFER AND LOOP DETECTION.....	17
4.6.1 <i>General.....</i>	17
4.6.2 <i>Internal and External Call Transfer.....</i>	17
4.6.3 <i>Call Transfer Loop Detection.....</i>	18
4.7 ANSWERING MACHINE DETECTION	19
4.7.1 <i>Introduction.....</i>	19
4.7.2 <i>How It Works.....</i>	20
4.7.3 <i>The Answering Machine Beep Tone Detection.....</i>	21
4.7.4 <i>The Answering Machine Detector Parameter</i>	21
4.7.5 <i>Typical Mobile Answering Machine Properties.....</i>	22
4.8 VOICE SERVER GROUPS AND REMOTE CONNECTIONS	23
4.9 FAULT TOLERANCE AND LOAD BALANCING	25
4.9.1 <i>Telephone Systems.....</i>	25
4.9.1.1 <i>Telephone Lines.....</i>	25
4.9.1.2 <i>LS1</i>	25
4.9.1.3 <i>FoIPv3</i>	27
4.9.2 <i>Voice Resource.....</i>	27
4.9.3 <i>Plug-ins</i>	28
4.9.3.1 <i>TC/VoiceAccess</i>	29
4.9.3.2 <i>TC/VoiceLink2</i>	29
4.9.3.3 <i>TC/VoiceXML</i>	29
4.10 SPEECH PROFILES	29
4.11 SERVICE PROVIDER INTERFACE – SPI	30
4.12 TC/VOICEACCESS.....	31
4.12.1 <i>Message Preprocessing and MetaMail Support (TC/PreProcess).....</i>	32
4.13 TC/VOICEXML	33
4.14 TC/VOICELINK2	34
4.14.1 <i>Polling</i>	37
4.14.2 <i>Response Handling</i>	38
4.14.3 <i>Templates</i>	39
4.14.4 <i>Multilingual</i>	39
4.14.5 <i>Types of Messages</i>	40
4.14.6 <i>User Properties</i>	40
4.14.7 <i>Text2Wave Pre-Conversion</i>	42
4.15 VOICEXSP	43
5. REQUIREMENTS	44
5.1 RECOMMENDED	44
5.2 ADDITIONAL REQUIREMENTS FOR ISDN	44
5.3 ADDITIONAL REQUIREMENTS FOR H.323 / SIP	44

5.4	ADDITIONAL REQUIREMENTS FOR TC/VOICEACCESS.....	44
5.4.1	<i>Recommended Number of Voice Channels</i>	44
5.5	ADDITIONAL REQUIREMENTS FOR TC/PREPROCESS	45
5.6	ADDITIONAL REQUIREMENTS FOR VL2	45
5.6.1	<i>Concurrent Voice Channels</i>	45
5.7	ADDITIONAL REQUIREMENTS FOR TC/VOICEXML	46
5.8	ADDITIONAL REQUIREMENTS FOR TC/SPI-LN	46
5.9	ADDITIONAL REQUIREMENTS FOR TC/SPI-MX	46
5.10	REQUIRED LAN BANDWIDTH	47
6.	LICENSING	49
7.	GENERAL MODULE INFORMATION	50
7.1	GENERAL	50
7.2	TC/VOICEACCESS.....	50
7.3	TC/VOICELINK2	50
7.3.1	<i>TCSI Logger</i>	50
7.3.2	<i>File Logger</i>	50
7.3.3	<i>TCSI Poller</i>	50
7.4	TC/VOICEXML	51
8.	INSTALLATION	52
8.1	WHERE TO INSTALL A VOICE SERVER.....	52
8.2	WHAT SHOULD NOT BE INSTALLED ON A VOICE SERVER	52
8.3	INSTALLING PREREQUISITES	52
8.3.1	<i>TC/SPI-LN</i>	52
8.3.2	<i>TC/SPI-MX</i>	52
8.3.2.1	Check Exchange 2010 Throttling Policies	54
8.3.2.2	Check Exchange 2013 Throttling Policies	54
8.3.2.3	Check Exchange 2016 Throttling Policies	54
8.3.2.4	Check MAPI Version Restrictions.....	55
8.3.2.5	Handling of Hidden Recipients.....	55
8.3.2.6	Additional Hints.....	56
8.3.3	<i>TTS</i>	56
8.3.4	<i>TC/VoiceLink2</i>	56
8.4	SETUP	57
8.4.1	<i>TC/VoiceAccess</i>	60
8.4.1.1	<i>TC/PreProcess</i>	62
8.4.1.2	<i>TC/VoiceAccess Continued</i>	63
8.4.1.3	<i>TC/SPI-MX</i>	66
8.4.1.4	<i>TC/SPI-LN</i>	69
8.4.1.5	<i>TC/VoiceAccess Continued</i>	69
8.4.2	<i>TC/VoiceLink2</i>	71
8.4.3	<i>TC/VoiceXML</i>	75
9.	CONFIGURATION	77
9.1	GENERAL	77
9.2	FIRST TIME CONFIGURATION	77
9.2.1	<i>TC/SPI-LN</i>	77
9.2.2	<i>TC/SPI-MX</i>	77
9.2.3	<i>TCECP</i>	78
9.2.3.1	<i>Plug-In Configuration</i>	78
9.2.4	<i>TCRT</i>	78
9.2.4.1	<i>Voice Server Groups</i>	78
9.2.5	<i>TC/VoiceAccess</i>	79
9.2.5.1	<i>Client Server Channel UC0</i>	79
9.2.5.2	<i>Services and Users</i>	79

9.2.5.3	Kofax Communication Server	82
9.2.6	<i>TC/PreProcess</i>	82
9.2.6.1	TCOSS Server and User	82
9.2.6.2	TFC License on TCOSS Server	83
9.2.7	<i>TC/VoiceLink2</i>	83
9.2.7.1	Setup, Users and Queues	83
9.2.7.2	The Kofax Communication Server	83
9.2.7.3	The Licenses	84
9.2.8	<i>VoiceXSP</i>	84
9.3	FEATURES	84
9.3.1	<i>TCECP</i>	84
9.3.1.1	TC/Mon32 Output	84
9.3.1.2	TRS.....	85
9.3.1.3	Delayed Active Plug-In Start.....	85
9.3.1.4	Maximum Memory Usage	85
9.3.1.5	Performance Counters.....	86
9.3.1.6	Timeouts	86
9.3.1.7	Graceful Shutdown	87
9.3.1.8	Dynamic Reconfiguration.....	88
9.3.1.9	Tracing	88
9.3.2	<i>TCRT</i>	89
9.3.2.1	Global TTS (De)Activation	89
9.3.2.2	TTS Engine and Speaker Configuration	89
9.3.2.3	Languages	91
9.3.2.4	Tracing	93
9.3.2.5	Location of the TCRT Engines	93
9.3.2.6	Playback of Remote Prompts	93
9.3.2.7	Audio Caching	94
9.3.2.8	SSML Support	96
9.3.2.9	System-Wide Default TTS Speaker	98
9.3.2.10	Voice Server Groups and Remote Connections.....	100
9.3.3	<i>TC/VoiceAccess</i>	101
9.3.3.1	The Attendant	101
9.3.3.2	Customized Greetings and System Prompts	102
9.3.3.3	User Depending Languages	103
9.3.3.4	Different Languages for Different Access Numbers.....	104
9.3.3.5	Alternative Operator	105
9.3.3.6	CallSender	105
9.3.3.7	Email Reader Support.....	107
9.3.3.8	Minimum Recording Length.....	107
9.3.3.9	Maximum Recording Length	108
9.3.3.10	Send a Fax with Voice.....	108
9.3.3.11	Dial by Name and Distribution Lists	109
9.3.3.12	Unique Single Number	110
9.3.3.13	Immediate Response.....	111
9.3.3.14	Customized Reply/Forward Header	111
9.3.3.15	Customizable Filename	112
9.3.3.16	Call Transfer and Loop Detection	112
9.3.3.17	Mailbox Sort Order	113
9.3.3.18	Sort Mailbox.....	114
9.3.3.19	Greeting Scheduler and Greeting Group	114
9.3.3.20	Importing Voice Prompts	121
9.3.3.21	Alternative Telephone Number and Representative	122
9.3.3.22	Disable/Enable “This message contains no attachments”	122
9.3.3.23	TCSI Codepage Conversion	122
9.3.3.24	Greeting Wizard	123
9.3.3.25	Lock Filter	123
9.3.3.26	Call Duration Logging	124
9.3.3.27	Enhanced Distinction of External and Internal Callers.....	124
9.3.3.28	Customized Subject	125
9.3.3.29	Account Locking	125

9.3.3.30	Password Expiry.....	126
9.3.3.31	Password Length	126
9.3.3.32	Tracing	126
9.3.3.33	Integration with Kofax Communication Server Archive	127
9.3.3.34	Deny Continuous Number for the PIN	127
9.3.3.35	Logging of Invalid Login Attempts.....	128
9.3.4	<i>TC/PreProcess</i>	128
9.3.4.2	Tracing.....	129
9.3.4.3	MetaMail Resolver	129
9.3.4.4	Watch Thread	132
9.3.4.5	Replacer.....	132
9.3.4.6	Converter	133
9.3.5	<i>TC/VoiceXML</i>	136
9.3.5.1	Built-In Grammars.....	136
9.3.5.2	Tracing.....	136
9.3.5.3	Defaults.....	137
9.3.5.4	Answering Machine Detection	138
9.3.5.5	Transfer the Voice Call to Fax.....	141
9.3.6	<i>TC/VoiceLink2</i>	142
9.3.6.1	Pollers	142
9.3.6.2	TSCI Poller – TCOSS Configuration.....	143
9.3.6.3	Telephone Number Prefixes	144
9.3.6.4	TCOSS Queue	144
9.3.6.5	Logging & Billing.....	144
9.3.6.6	Remote Voice Server	146
9.3.6.7	Answering Machine Detection	146
9.3.6.8	Individual IVR Plug-in	147
9.3.6.9	Address Syntax Customization	147
9.3.6.10	TCSI Codepage Conversion	148
9.3.6.11	Recipient Properties	149
9.3.6.12	Poll Cycle	149
9.3.6.13	Timeouts.....	150
9.3.6.14	Enhanced Message Termination.....	151
9.3.6.15	Disable TTS	154
9.3.6.16	Tracing	155
9.3.6.17	Directories Used	155
9.3.6.18	Base URL	156
9.3.6.19	Old VoiceLink Compatibility	156
9.3.6.20	Default Template.....	157
9.3.6.21	Message Properties.....	157
9.3.6.22	Keep Remainings	158
9.3.6.23	Placeholder Identification.....	159
9.3.6.24	Customized Send Options	159
9.3.6.25	Number Locking	161
9.3.6.26	Text2Wave Pre-Conversion	162
9.3.6.27	Customizing Audio Extensions	163
9.3.7	<i>TC/VRedirect</i>	163
9.3.7.1	Caching Of CustomerIDs.....	163
9.3.7.2	Service of a Voice Call	164
9.3.7.3	Tracing	164
10.	SCENARIOS AND TUTORIALS	165
10.1	TC-PROD-VOICE – TYPICAL INSTALLATION	165
10.1.1	<i>The Script</i>	165
10.1.2	<i>The Actors</i>	165
10.1.3	<i>The Setting</i>	166
10.2	THE xSP – SERVICE PROVIDER	166
10.2.1	<i>The Script</i>	166
10.2.2	<i>The Actors</i>	166
10.2.3	<i>The Setting</i>	167

10.3	VOICEXML	168
10.3.1	<i>The Script</i>	168
10.3.2	<i>The Actors</i>	168
10.3.3	<i>The Setting</i>	168
10.4	TC/VOICELINK2	168
10.4.1	<i>Customizing the TSCI Poller's Termination Behavior</i>	168
10.5	TC/VOICELINK2 – TRANSFORMATION OF A MESSAGE	169
10.5.1	<i>Writing a Template</i>	169
10.5.2	<i>The URL Message</i>	172
10.5.3	<i>The VoiceXML Message</i>	172
10.5.4	<i>The Text Message</i>	173
10.5.5	<i>The User Properties Message</i>	173
10.6	TC/SPI-LN MANUAL INSTALLATION	174
10.7	TESTING IVR WITHOUT A TELEPHONE SYSTEM.....	174
10.8	REMOVING THE OLD TC/VOICELINK.....	175
10.9	REMOVING THE TC/VOICELINK2 VERSION 1.0 AND 1.2	175
10.10	FAULT TOLERANT TC/VOICELINK 2.2 INSTALLATION.....	175
11.	SECURITY	177
11.1	USED IP PORTS	177
12.	AUDIO FORMATS	178
12.1	GSM 610 SUPPORT FOR CISCO NATIVE INTEGRATION	179
13.	SUPPORTED TTS ENGINES	180
14.	TELEPHONE SYSTEMS	181
15.	PERFORMANCE	182
16.	KNOWN ISSUES	184
16.1	RESTRICTIONS.....	184
16.1.1	<i>General</i>	184
16.1.2	<i>TCRT</i>	184
16.1.3	<i>TTS</i>	184
16.1.4	<i>TC/VoiceAccess</i>	185
16.1.4.1	<i>TC/PreProcess</i>	185
16.1.5	<i>TC/VoiceLink2</i>	185
16.1.6	<i>TC/SPI-LN</i>	186
16.1.7	<i>TC/SPI-MX</i>	186
16.1.8	<i>TTS Conversion Caching</i>	187
16.2	TROUBLESHOOTING	187
16.2.1	<i>Corrupt TTS output with ELAN TTS 4.6</i>	187
16.2.2	<i>TC/SPI-LN</i>	187
16.2.3	<i>Acapela TTS</i>	188
16.2.4	<i>The VL2 Does Not Play Umlauts When Using Acapela TTS</i>	188
16.2.5	<i>TC/VoiceLink2</i>	188
16.2.6	<i>Fault Tolerant Voice</i>	188
16.2.7	<i>Bad TTS Audio Quality</i>	188
17.	HINTS	189
17.1	TC/VOICEACCESS.....	189
17.1.1	<i>TC/PreProcess</i>	189
17.2	TC/VOICELINK2	189
17.3	TCRT	190

17.4	TTS.....	190
18.	UPGRADE OR DOWNGRADE.....	191
18.1	FROM VL2 1.0/1.1 TO VL2 1.2 (AKA VL2.2).....	191
18.2	FROM VL TO VL2	192
18.3	FROM SAPI 4 TO SAPI 5.....	193
19.	DOCUMENTATION CHANGES	195
19.1	7.80.00 EDITION 5 -> 7.80.04.....	195
20.	THIRD PARTY LICENSES	196
21.	REFERENCED DOCUMENTATION.....	197
22.	TERMINOLOGY	198
22.1	ABBREVIATIONS	198
22.2	GLOSSARY	199
23.	CONFIGURATION PARAMETER OVERVIEW.....	201
23.1	GENERAL	201
23.2	TC/PREPROCESS	201
23.3	TCE_RES	202
23.4	TCE_Rem	203
24.	APPENDIX A – INDEX	206
25.	APPENDIX B – TABLES AND FIGURES.....	208
25.1	TABLES	208
25.2	FIGURES.....	208

1. How to Read This Document

The document and each chapter are written in a top down approach, starting with the basics and going into details later. If basic knowledge is already at hand, the first chapters about the [Fundamental Definitions](#) and the [Concepts and Architectures](#) can be skipped but may be worth to skim through.

This document is designed for the technician and the interested administrator to find the requested information about configuration and installation as fast as possible.

It is not designed as a programmer's reference.

References to chapters are realized as hyper links. The references are made directly within the sentence where the name of the chapter is used. E.g.: Please read [How to Read This Document](#) for information about how to read this document.

Numbers in square brackets like [\[10\]](#) are references to other documents. These references are listed in the chapter [Referenced Documentation](#).

When referencing a registry key, only the relevant parts are written. This means the key "HKLM\Software\TOPCALL\TCECP\TCRT\TTS\Interface" would only be written as "TCRT\TTS\Interface".

2. Introduction

The Voice Platform is not only a voice mail system. The Voice Platform has been designed to allow telephone call handling in almost all thinkable ways. Incoming or outgoing calls, transfer calls, pre-recorded messages that are stored either locally or on a web server, or that contain generated text written in different languages which need to be synthesized by different TTS engines.

The Voice Platform is fault tolerant and load balanced.

Although several modules are already shipped with the Voice Platform, the Voice Platform can be easily enhanced.

At the moment, the following applications are shipped with the Voice Platform:

- TC/VoiceAccess which is the voice mail application with a basic attendant.
- TC/VoiceLink2 which can be used to send automatic outbound voice calls.
- TC/VoiceXML is a framework to run VoiceXML applications.
- TC/VRedirect used for VoiceXSP scenarios.

The Voice Platform is the solution when it comes to connecting telephone with business.

3. Fundamental Definitions

Please read this chapter for a short introduction to the terminology of the Voice Platform.

The **Voice Server** is another name for the Voice Platform. In this manual, the Voice Server abbreviation VS will be used instead of the term Voice Platform.

The expression “VS” contains two possible meanings:

- 1.) It describes a windows server where one TCECP process is running.
- 2.) It describes one instance of TCECP.

The difference is subtle and will get clearer when reading the chapter about VoiceXSP and fault tolerance. If not otherwise stated, definition one is used when talking about a VS.

To avoid confusion, please note the difference between: **channel**, **session**, **instance** and **call**.

Whereas in most cases, a channel will be equivalent to a session, it is important to know the difference.

A channel is the logical connection between the telephone system and the VS. When a user places a call, one channel from the PBX to the LS1 is used to establish the connection. This means that each call needs at least one channel. For every call, the VS will start a new session which will be the container for a new instance of a module. It is possible that the call will be transferred to a different destination by a plug-in. In this case, a second channel will be required. It is also possible that during a session, the responsible plug-in is changed. In this case, the old instance will be destroyed and a new plug-in instance will be created.

Briefly described, this means:

- A **call** refers to the activity needed to interact with a user.
- A **channel** is the logical connection between the VS and the telephone system required to place a call.
- A **session** is a container allowing plug-ins to handle incoming calls or place outgoing calls. A session might also be active when no call is active.
- An **instance** is the part of a plug-in which is running in the context of a session. Each session must have exactly one instance of a plug-in running.

Unfortunately, the term “instance” is also used when referring to VS instances. Depending on the context, either the one or the other is meant.

4. Concepts and Architectures

4.1 Overview

This chapter gives a basic overview about the architecture, the concept and the involved components. The following chapters will discuss certain parts in details.

The Voice Server architecture was designed with a high level of abstraction in mind. This allows components to work mostly independent from other components. The result was a three layered architecture.

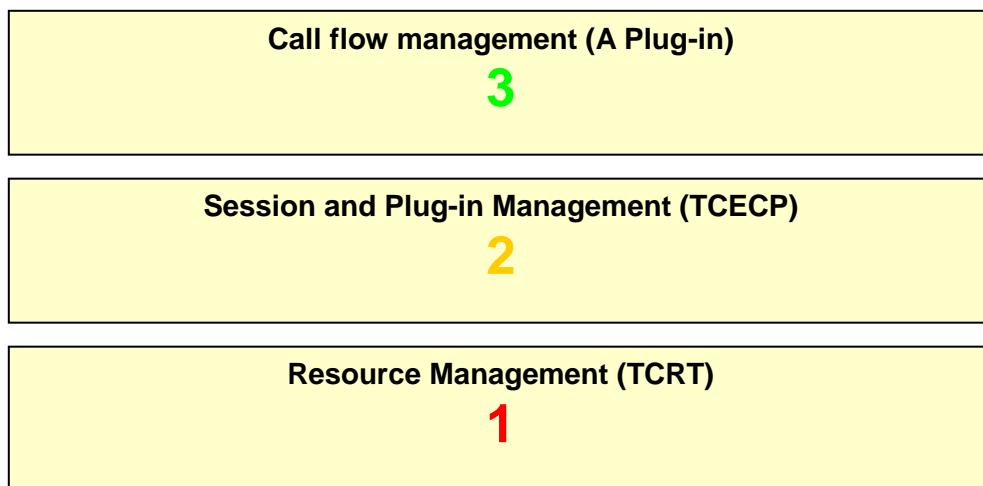


Figure 1 – The Three Layer Architecture

The **first layer** is the resource management layer. All the low level processes like knowing ISDN protocols and managing audio files are done here. It is the domain of TCRT.

The **second layer** is handled by TCECP, which creates and destroys sessions and decides which plug-in should handle a call.

The **third layer** is the place where a plug-in controls what the caller hears and what will happen next.

This three layer architecture leads to the following big picture (Figure 2 – The Voice Architecture in Details):

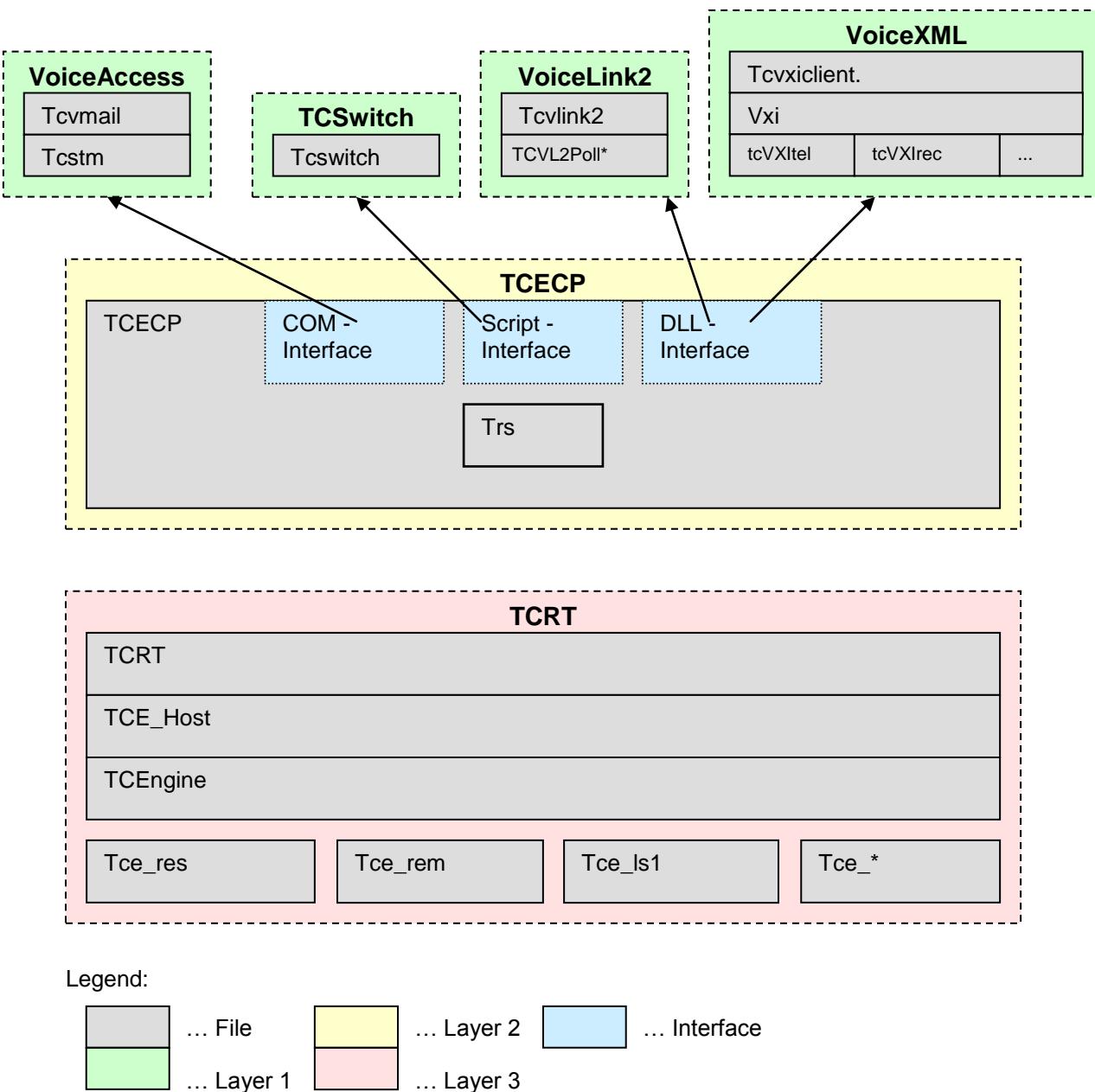
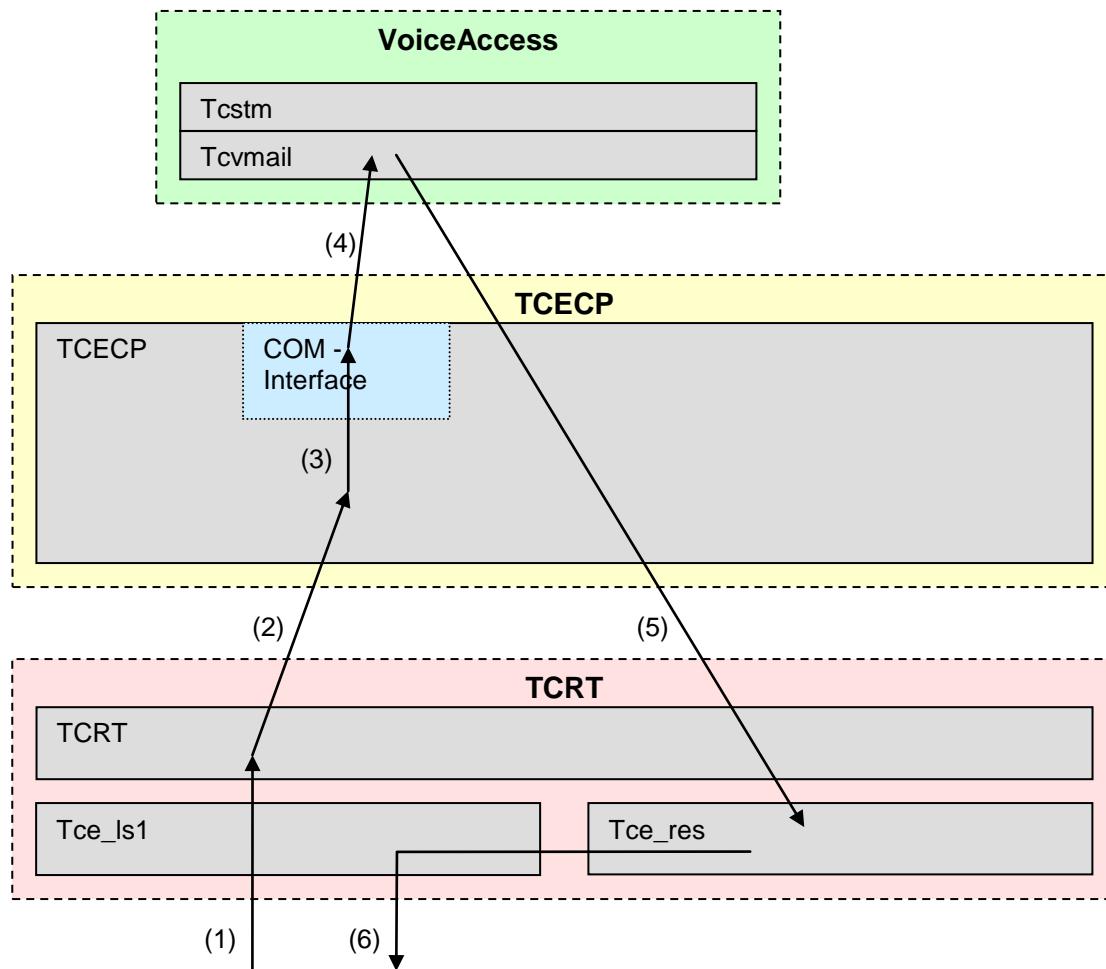
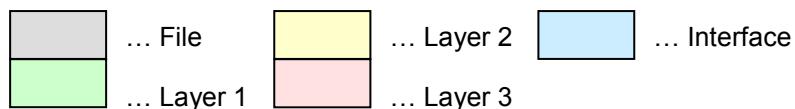


Figure 2 – The Voice Architecture in Details

A typical workflow of an incoming call will look like shown in Figure 3 – A Simple Call Flow:



Legend:

**Figure 3 – A Simple Call Flow**

If a call arrives, it will be received by the telephone engine of TCRT (1). TCRT will then notify TCECP of the incoming call (2). Next, TCECP initiates the plug-in which is responsible for calls and creates a new session (3). At last, TCECP passes the call control over to the plug-in (4). The plug-in will then perform some actions. For example, it will tell TCRT to play a prompt (5) to the caller (6).

4.2 The Concept of Plug-ins

Where TCECP and TCRT provide the framework, plug-ins are responsible for the actual call processing. The plug-in triggers playback, recording, call transfer and all the other features provided by the VS.

One main concept for the plug-in interfaces is the “Application – Instance” approach. When TCECP starts, it creates an application object of each plug-in (with the exception of plug-ins of type SCRIPT1). When an incoming call arrives, TCECP creates an instance of the plug-in which is to handle this call and will pass the application object to the new instance.

This has some advantages. First, this saves obsolete repetitions like entering and constantly having to re-enter configuration settings. Second, it is possible to share data between instances.

TCECP offers four different interfaces for plug-ins.

- 1) simple scripting interface (SCRIPT1)¹
- 2) advanced scripting interface (SCRIPT2)
- 3) COM interface (COM1)
- 4) DLL function interface (DLL1)

SCRIPT1 is the most basic plug-in interface. There is only one function required and was designed for fast prototyping and call processing without user interaction. (e.g.: TC/Switch) The script must be written for the MS-Scripting Host

SCRIPT2 is a more advanced interface for scripts. Like COM1 it offers the possibility of an application object which is passed to every instance. It can be used for basic TUI or more complex call processing. Like SCRIPT1, this script must be written for the MS-Scripting Host.

COM1 is the old standard interface for plug-ins. As the name suggests it uses COM – Objects which can be written in any language. It is/was used by VA. This interface offers the possibility of an application object and can be used for complex TUI applications.

DLL1 is the latest interface. It offers the most features (e.g.: LSR, Dynamic Reconfiguration ...). This interface is a mix of plain old C-functions and C++ objects. It can be used for prototyping and also for complex TUI applications.

4.2.1 Active Vs Passive Plug-in

There are two additional distinctions of plug-ins. A plug-in can be active or passive.

The difference shows whenever a new instance of the plug-in is created.

A passive plug-in waits for an incoming call. A new instance will be created when a call arrives for this plug-in. A typical example is VA.

An active plug-in does not wait for an incoming call. TCECP starts a defined number of instances during start up without an active call available. A typical example is the VL2.

Attention: A plug-in cannot be active and passive at the same time.

4.3 Aliases

With the VS every plug-in can have multiple names identifying a plug-in.

E.g.: VA may be called “VoiceMail”. TC/VoiceXML may be called “VoiceXML”, “Attendant” and “Login”

¹ This interface does not support the “Application – Instance” approach.

This allows an easy replacement of plug-ins with as minor changes as possible.

Here's a short example: Let's assume TC/VoiceXML was configured as "Attendant" and responsible for all calls with a called ID of 99. Now a new plug-in should do the job of the "Attendant" instead. The only thing that has to be done is to remove the alias "Attendant" from TC/VoiceXML and add it to the new plug-in.

Further does the alias allow plug-ins to use the LSR without knowing the type or real name of the destination plug-in. It simply tells the VS to start "VoiceMail" regardless if this will result in starting VA or a different plug-in.

Each alias is also associated with a display name. This name is displayed in TC/Mon32.

4.4 TRS – Transparent Routing Script

The idea behind the TRS is to have a flexible way to tell the VS which plug-in shall be started when a call arrives. This is no trivial task because there may be different plug-ins with different responsibilities.

It may be necessary that every call to a specific extension is handled by one plug-in but calls to this extension from a special caller must be handled by a different plug-in.

To provide this maximum flexibility the TRS was installed. It has an interface which is quite similar to the SCRIPT2 interface and also supports the "application – instance" approach.

The VS does initialize the TRS as one of the first things. This allows the TRS to prepare parts of the environment in advance. As an example the TRS can extract the CustomerID from TCOSS and modify the registry of TCRT to allow customer based routing in layer 3.

After initialization, the TRS is activated for every incoming call. The call is passed to the TRS which decides which alias should handle the call. Note that it does not show the plug-in name but the alias. To find a decision the TRS can use all information provided by TCRT itself and by the MS-Scripting Host. Based on the alias returned by the TRS the VS creates a new instance of a plug-in.

In short: The TRS is invoked for every incoming call but only to determine the responsible alias.

4.5 LSR – Local Session Redirection

There is no connection between the TRS and the LSR. The LSR enables plug-ins of type DLL1 to pass control to another plug-in of type DLL1.

A typical scenario is the VL2. The VL2 is only responsible to establish the call to the recipient. When the connection is established, the VL2 redirects the session to the plug-in with the alias VoiceXML. The VS will now create a new instance of "VoiceXML" and pass the call to it. The old instance of the VL2 is destroyed. Later when VoiceXML is done (e.g.: because the recipient hung up), the VS reverses the procedure. Now a new instance of the VL2 is created and destroys the VoiceXML instance. The VL2 can process the response or do whatever has to be done.

Its behavior is like a function call stack.

4.6 Call Transfer and Loop Detection

4.6.1 General

If using one of the functions Call Sender, Attendant or Dial by Name of VA, or “<transfer>” of VoiceXML the VS has to transfer the caller to the new destination number (like operator, the originator of a voice message etc.). In order to do so, the VS first establishes the 2nd call to the desired new destination and invokes the Call Transfer function that interconnects the caller with the new destination number.

Note: This functionality is not related to LSR.

It is clear that the most important task is to provide the supported functionality, but on the other hand attention must be paid to avoid unnecessary call loops between the VS and the PBX that could easily occur with more complex Call Transfer scenarios.

This chapter explains which kind of call loops may occur and how they can be optimized.

For a more detailed technical background please refer to the ISDN Technical Manual [\[3\]](#).

4.6.2 Internal and External Call Transfer

- 1) Internal Call Transfer means that the VS (being the „User A”) switches two of its calls internally using so called *tromboning* or *bridged transfer*. The disadvantage of *tromboning* is that two lines towards PABX are occupied for one caller:

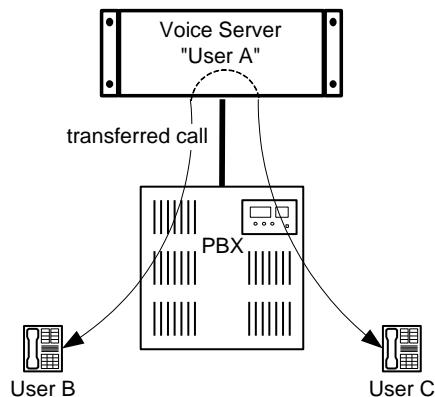


Figure 4 – Internal Call Transfer

On the other hand the advantage is that it works with any type of LS1 PBX integration.
It is not supported by the CISCO integration.

- 1) External Call Transfer means that the Voice server („User A”) transfers its 1st caller (User B) to new destination number (User C) externally via the PBX. This technique is called *blind transfer*.

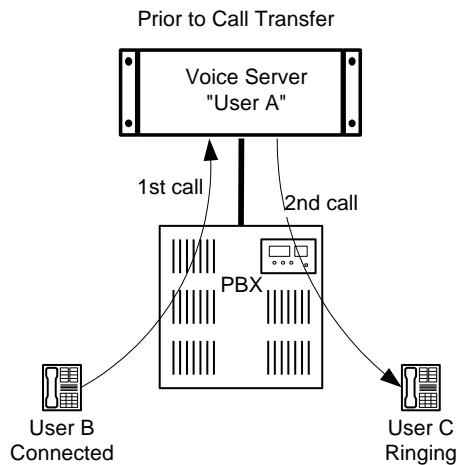


Figure 5 – External Call Transfer Prior to the Transfer

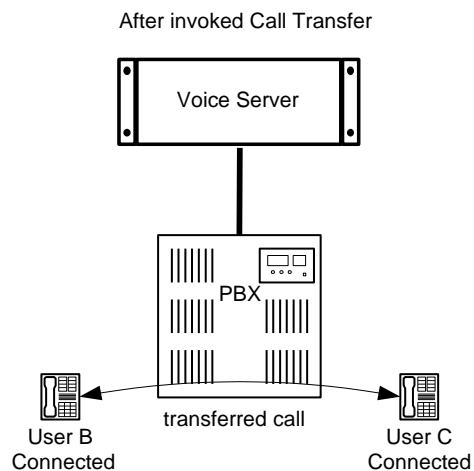


Figure 6 – External Call Transfer After the Transfer

The advantage is that no lines are occupied between Voice Server and the PABX while users B and C are talking.

Please refer to the ISDN Technical Manual for the prerequisites for the External Call Transfer (LS1 and PBX configuration).

If the external Call Transfer is requested but not supported by the particular LS1 PBX integration, an internal Call Transfer will be used automatically instead.

4.6.3 Call Transfer Loop Detection

External Call Transfer (via the PBX) works fine if the User C to whom the User B is being transferred to, answers the call. In this case the PBX optimizes the connections and interconnects users B and C internally.

But if the User C does not answer the call and his phone has an activated Call forward function that forwards all his calls to the Voice Server (if any of call forward conditions like busy, not responding or unconditional occurs) the PBX will establish the 3rd call towards Voice Server:

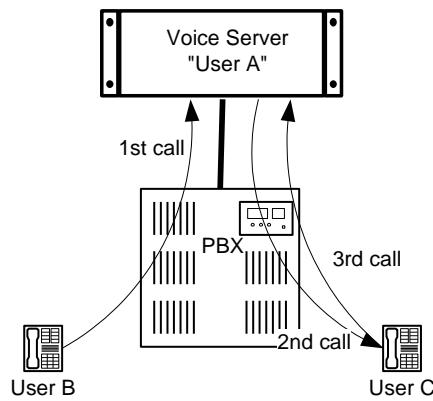


Figure 7 – Call Transfer Loop Detection

Thus the caller (User B) will be connected with User C's mailbox via 3 lines.

Note that even if the LS1 PBX integration supported the External Call Transfer, such a complex looping would not be optimized. The caller will occupy 3 lines until leaving a message for User C or even more if he chooses to be transferred to any other user (e.g. by using Dial by name function).

However, the Voice Server recognizes that the 2nd call of the Call Transfer is being forwarded back to the own server. The basic requirement for this is that all extensions in the company must be forwarded to the same company wide Voice Access Number.

Please refer to the chapter “Call Transfer Loop Detection” in the ISDN Technical Manual for prerequisites and configuration of the LS1 server.

TC/VoiceAccess supports Call Transfer Loop Detection without any additional configuration burden.

Please refer to chapters “Dial by Name” and Call Sender” for details on optional configuration changes.

4.7 Answering Machine Detection

4.7.1 Introduction

The VL2 is often being used for various automatic alerting or notification tasks by playing voice or text messages to the user's telephones or mobiles. As soon as the called person has answered the call, he/she is usually informed that the call has been originated by an automatic message generation system and is prompted to acknowledge the receipt of the call/message by pressing any pre-defined button on the phone. The VL2 waits until the callee does so and if he/she does not, it will repeat the corresponding prompt for a couple of times. If the callee still does not press a button, the VL2 will end the line and can (configuration dependant) call again later through retries.

Often, the called person is busy or not responding and the VL2 call is diverted to his/her voice mailbox or answering machine. The answering machine typically plays a greeting saying that the callee is not reachable and invites the caller to leave a message after the beep tone. If VL2 did not recognize that the answering machine answered the call, it would fill up the mailbox by useless messages prompting the callee to press a button to listen to the message etc.

VL2 recognizes that the call was diverted to the answering machine and handle it in an appropriate way, for example finish the call or record a dedicated message to the user's mailbox.

4.7.2 How It Works

Nowadays the advanced DSP (Digital Signal Processing) technology makes answering machine/mailbox detection possible. There are different methods to accomplish this task:

- 1) Analyzing the energy and the cadence of an incoming signal: typically the human responding the call under his private telephone number would say something like "Hello" or „Here is Fred Flintstone" shortly (typically up to 2 – 2.5 seconds) and then wait for the caller to start talking.

So the possible criterion for the answering machine detector may be the sequence of 2 seconds voice and 2-3 seconds silence afterwards (on the other hand, the answering machine would typically talk longer than 2 – 2.5 seconds)

- 2) The cross-talk detection is based on the fact that during a telephone conversation, two humans usually talk in a half-duplex manner (if they do not argue or shout at each other) one speaks while the other one listens. On the other hand, in the case of two machines conversation (e.g. TC/VoiceLink and the answering machine); there would be a period at the beginning of the conversation where both are talking at the same time.

So the possible criterion for the answering machine detector may be the cross-talk detection for about 2-3 seconds at the beginning of the call.

- 3) The beep tone detection: nowadays almost every answering machine/voice mailbox indicates the beginning of the recording by a clear single frequency beep tone, only very few of them seem to use DTMF-like dual-frequency tones.

Each of these methods has its own advantages but also drawbacks:

Method 1) and 2) are likely to fail if the callee's greeting on the mailbox is too short: tests with real-world mailboxes show that their lengths vary from about 2,5 – 3 seconds up to 30-40 seconds so it is very difficult to distinguish between a real human who has just said "hello" and the extremely short mailbox greeting where only the callee's name has been recorded without any other announcement. They would also tend to fail in an extremely noisy environment (like in the car etc.) where the energy of the silent part may be as high as real voice due to the high level of background noise.

On the other hand, the advantage of methods 1) and 2) may be the fact that the answering machine would be recognized very quickly (after 2-3 seconds). Therefore such methods are often used by different telemarketing companies attempting to optimize the time of their call agents: they call the telephone number by an automatic call server and as a human being is recognized having answered the call, the call is transferred to the next free agent.

The advantage of method 3) is that it does not suffer under noisy environment as the single beep tones are clearly distinguishable from noise through the spectral analysis. Also it does not depend on the minimum length of the mailbox welcome prompt.

The disadvantage of the method 3) may be the fact that the answering machine is detected quite late (not unless the beep tone has been played) which makes this method unusable for telemarketing campaigns as the system would wait too long for the beep tone and the callee would quit before being transferred.

4.7.3 The Answering Machine Beep Tone Detection

As VL2 is not used for telemarketing campaigns, the only criterion is to recognize the answering machine with a sufficient recognition rate and to provide very low rate of false-detects (the real human recognized as the answering machine).

There are two methods of operation:

1) Auto-detection mode

The detection criterion of this method is the sequence of:

- configurable silence period prior to the beep tone (default 16 ms)
- single tone frequency in the range 350 Hz – 3400 Hz of a minimum duration (typically 100ms)
- configurable silence period after the beep tone (default 300ms)

This method is suitable to detect virtually all mailboxes using the single frequency beep tone.

2) Restricted detection mode

Uses the same sequence of events (silence, minimum length of the beep tone, silence) as the auto-detection mode with the difference that up to 8 distinct frequencies in the range 350Hz – 3000Hz can be specified (and only they would trigger the event beep detected).

This mode works for xSP deployment model where the service provider would intend to intentionally recognize only own answering machine while ignoring the others.

However, the frequency of the beep tone being used must be well-known and must be configured at first.

Answering machine detection test results:		
	Mailbox detection rate	False mailbox detection rate
Auto-detection mode (default settings)	95% (of 130 real-world mailboxes)	0.02% (of 100.000 non-mailbox voice recordings)

Table 1 – Answering Machine Detection Test Results

4.7.4 The Answering Machine Detector Parameter

Generally, answering machine detection works on a call-by-call basis on the DSP resources on the LS1, it is activated by the VL2 just after the particular call has been connected through.

The properties for the detector are set by the means of a single configuration string. The syntax of the configuration string is:

```

"DetTime=<ms>[,StartDelay=<ms>,Toff1=<ms>,Toff2=<ms>,BlkLen=<int>,DetEnergy=<int>,
[Fq1=<Hz>,Fq2=<Hz>,OnTime=<ms>], 
[Fq1=<Hz>,Fq2=<Hz>,OnTime=<ms>],
...
[Fq1=<Hz>,Fq2=<Hz>,OnTime=<ms>]] /* up to 8 Fq1/Fq2 parameters */

<ms> - Time in milliseconds
<Hz> - Frequency in Hertz
<int> - General integer value

```

At the moment, only the VL2 supports mailbox detection. See [VL2 - Answering Machine Detection](#)

The detailed detector properties description:

Property	Default value	Description
DetTime	0 ms	The maximum detection time in milliseconds, if this timer expires the detector will be stopped for the particular call. Setting DetTime to 0 disables detection for the particular call
StartDelay	2000 ms	The detection starts StartDelay milliseconds after being activated in order to ignore possible call progress tones or their rests just after the call has been put through (When the call is being diverted to the mailbox, some providers indicate this by a short ring tone at the beginning)
Toff1	16 ms	The minimum length of silence in milliseconds just in front of the beep tone. Setting Toff1 != 0 improves false detects on non-mailbox recordings
Toff2	300 ms	The minimum length of silence in milliseconds after the beep tone. Setting Toff2 != 0 improves false detects on non-mailbox recordings, but setting it too high may cause some messages being recorded by the mailbox as the call would be disconnected too late
BlkLen	128 or 160 samples	The block length in samples the detector operates on: a.) Auto mode – 128 samples => 16 ms b.) Fixed mode – 160 samples => 20 ms
DetEnergy	80 %	The minimum ratio of single tone energy in % of total signal energy as a criterion for the tone to be recognized as the “beep tone”
Fq1 and Fq2 ^(*)	-1 or >0	Setting of the frequency of the desired beep tone(s): a.) Fq1=-1 ... Auto mode (no explicit frequency configuration) b.) Fq1=<Hz> ... Fixed frequency in Hertz Property Fq2 must be always 0, is reserved for future usage
OnTime ^(*)	100ms	The minimum length of the tone in milliseconds

Table 2 – Answering Machine Detection Configuration Parameters

(*) – the triple Fq1, Fq2 and OnTime may be set up for max. of 8 times for restricted detection mode but only once for auto-mode

Activating the detector by this simple command string

```
"DetTime=60000"
```

is equal to the full (default) format:

```
"DetTime=60000,StartDelay=2000,Toff1=16,Toff2=300,BlkLen=128,DetEnergy=80,
Fq1=-1,Fq2=0,OnTime=100"
```

4.7.5 Typical Mobile Answering Machine Properties

Nowadays, the vast majority of mobile providers and answering machine vendors also tend to use the single frequency beep tone in order to indicate the beginning of the recording. Only very few answering machines appear to use the dual-frequency tones (<1 %).

The following table summarizes experimental results of testing with a large amount of different mailboxes during spring 2006. Note that there is no mandatory standard for the characteristics of the beep tones being used in the field and therefore, mobile providers may change them anytime (for example, the Austrian provider Telering used dual-frequency tone in 2005 but changed to 1400 Hz single tone as the majority of the others do).

It looks like that the global providers (like T-Mobile and Vodafone) use the same tone in (at least) the most of countries they are present.

Many mobile providers offer also the possibility to leave a message directly to the mailbox without even ringing the person's mobile phone: while in Austria and Germany this works usually by inserting a 2 digit prefix between the provider area code and the mobile number, many other providers offer this function by dialing a special service number and then typing the mobile number after being instructed by the operator's voice (2 stage dialing). Please find instructions how to leave the message directly for the particular provider in the following table. This function is of course not essential for the answering machine detector but it offers an excellent possibility to perform a couple of test, without disturbing anyone:

Provider	Country	Beep Tone Frequency [Hz]	Beep Tone Duration [ms]	Direct mailbox access (for the hypothetical number 123456)
T-Mobile	Austria	736	128	0676- 22 -123456
	Germany			0170- 13 -123456
	Slovakia			1. stage: 0903-333-666 2. stage: 903-123456 (2 nd without '0' !)
Vodafone	Hungary	736	128	(*)
	Spain			(*)
	Germany			0170- 50 -123456
	Sweden			1. stage: 070-6133-100 2. stage: 070-123456#
One	Austria	736	128	0699- 3 -23456 (exchange the 1. digit '1' for '3')
A1	Austria	1400	340	0664- 77 -123456
Telia Sonera		1400	340	(*)
SwissCom	Switzerland	1400	340	086-079-123456 (086 is the special area code for direct mailbox access)
Telering	Austria	1400	250	(*)
Orange	Slovakia	800	500	(*)

Table 3 – Answering Machine Detection Frequencies and Durations

(*) – Direct mailbox access is unknown for this provider

4.8 Voice Server Groups and Remote Connections

There is one thing Voice Server Groups [VSG] and remote connections have in common, the remote connection.

A remote connection is any connection to a VS initiated to use a resource on the VS.

A resource might be the telephone engine or a plug-in. E.g., if a user tries to play/record a message with the TC/Player over the telephone, a remote connection will be established from the TC/Player to a VS. The TC/Player then uses the telephone resource on the VS to play/record the message.

Another example is the VL2. A VL2 can run on a VS without its own telephone engine. To perform outgoing calls, the VL2 establishes a remote connection to a VS using a telephone engine.

The last example is part of the VoiceXSP scenario. In this example, a VS receives an incoming call. The VS decides who (VS) and what (Plug-in) is responsible for this call. Then the VS creates a remote connection to the VS with the desired plug-in.

TCRT automatically exposes the resources of the available engines but it is also possible to add so called fake resources.

The VSG and their relations to remote connections:

For large installations, it might be necessary to deploy more than only one VS. For very large installations, it is advised to restrict the usage of certain voice servers to a certain group of users. (see Figure 8 – Multiple Voice Server Groups)

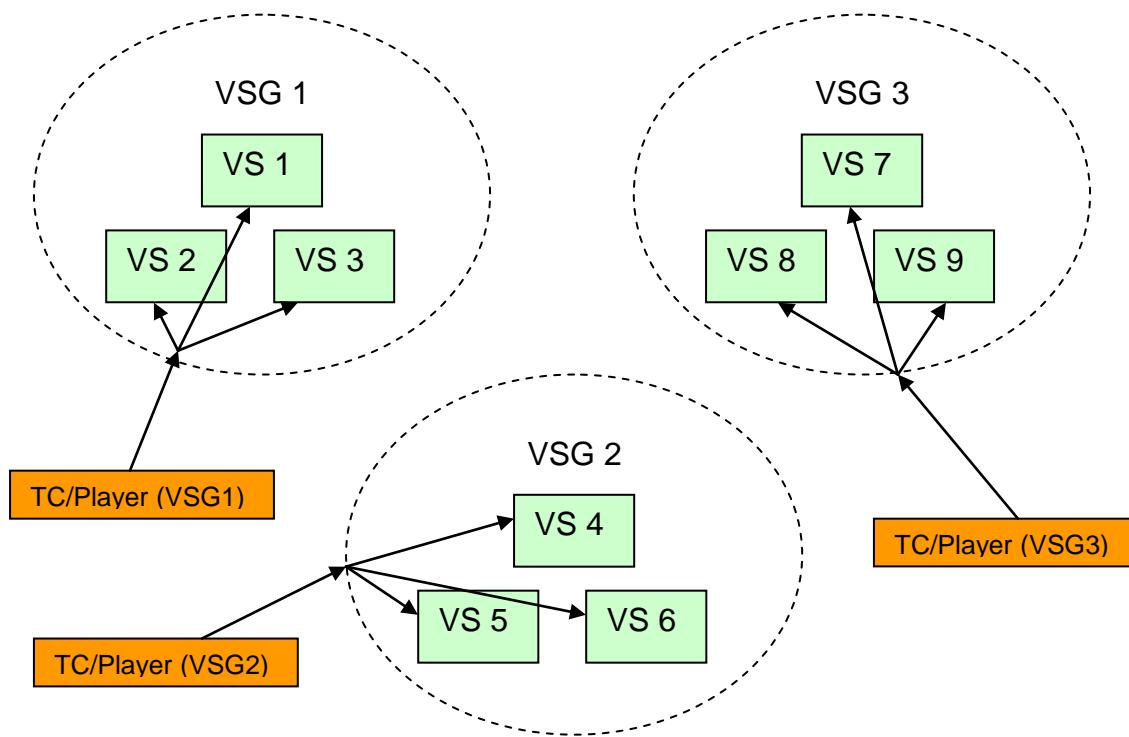


Figure 8 – Multiple Voice Server Groups

To avoid repeating VS configurations for every client (TC/Player) the VSG was designed. It is possible to put multiple voice servers into one group and have multiple VSG in a network.

The administrator only tells the client which VSG to use and the rest will be done automatically.

The client will automatically connect to the next free VS in the VSG.

The client will automatically detect new VS in the VSG.

The client will automatically fail over to the next VS if the current is does not respond any more.

It is still possible to use the VS of a different group by telling the client exactly which VS to use.

Please see the [Fault Tolerance and Load Balancing](#) chapter for more information.

4.9 Fault Tolerance and Load Balancing

Fault tolerance is the ability of the system to continue its operation even in the case of unexpected hardware or software failure(s). The Kofax approach is to guarantee the functionality in the case of any *single component failure*. If multiple faults occur at the same time, the overall operation of the system cannot be guaranteed.

Load balancing is the method to distribute the call traffic evenly on the network of available resources so that no single resource is overloaded.

An overview of fault tolerance can be found here [\[3\]](#).

This chapter will try to explain all the different possibilities of load balancing and fault tolerance by splitting it into the parts:

- Telephone systems
- Voice Resources
- Plug-ins

4.9.1 Telephone Systems

4.9.1.1 Telephone Lines

The LS1 server provides the ISDN interface to the PSTN (Public Switched Telephone Network) which can be deployed by the means of either public ISDN lines connected to the CO (Central Office) or by going across the PBX (Private Branch Exchange).

In order to provide fault tolerance and load balancing, two or more LS1 servers should be connected to the PSTN (or PBX) using at least two separate ISDN lines:

- The incoming call distribution towards LS1 must be provided by the CO or the PBX by assigning one range of telephone number for a group of more than one physical ISDN lines. Thus, the switch can distribute the calls evenly across all connected ISDN lines (load balancing), but provides also fault tolerance by skipping lines which are out of service (LS1 malfunction, line disconnected etc.)
- The outgoing call distribution from the LS1 towards the telephone network is provided by each LS1 server itself for all ISDN lines that are connected to it: the LS1 distributes all outgoing calls evenly to the ISDN lines in service and skips those out of service.

In the case the LS1 receives an outgoing call request from any application but all its lines are busy (or out of service), it reports immediately and further retries will be performed by the means of [Voice Resource](#) based fault tolerance and load balancing.

4.9.1.2 LS1

Each voice server can interconnect with multiple LS1 servers by the means of running multiple instances of TCECP on the same machine connecting to different LS1 with different priorities. The LS1 engine contains a parameter for the connection to a LS1 which is called connection priority. The LS1 will disconnect a connection with lower priority if another connection with higher priority is attempted. A connection attempt is refused if the LS1 already has a connection with higher priority. Please see Figure 9 for an example.

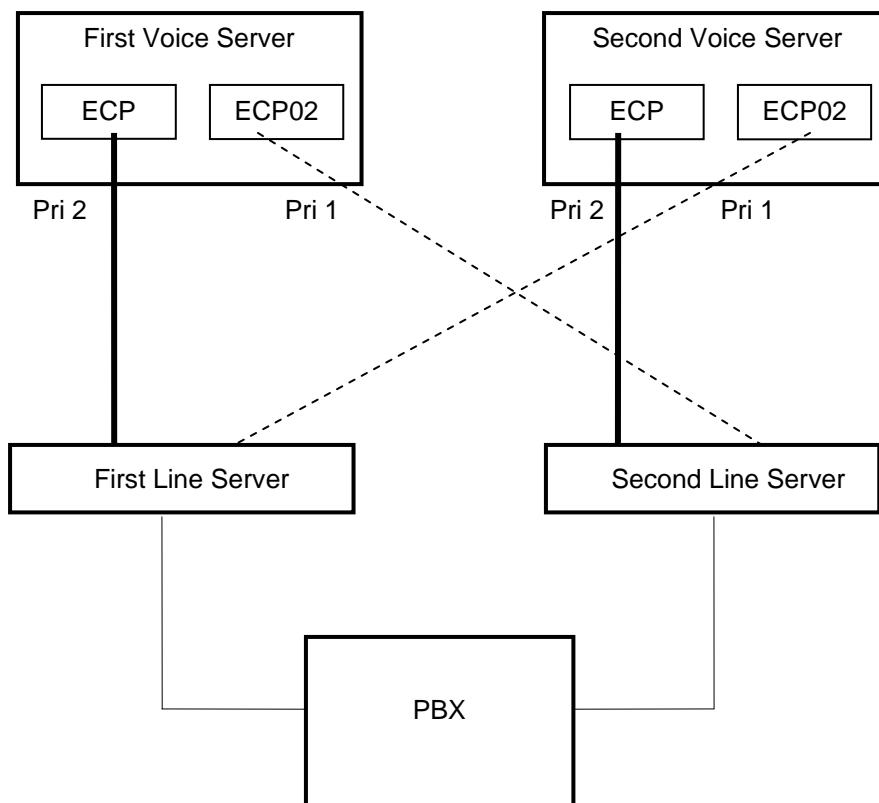


Figure 9 – System in Full Fault Tolerant State. Both Voice Servers Are Running

Note: First server and second server here do not necessarily mean a TCOSS Model 22x Tandem systems hardware, since it is not recommended to install TCECP on the same machine as TCOSS. It would rather be two independent voice servers accessing the same Kofax Communication Server running on a different hardware which can be a Model 22x Tandem system. The Kofax Communication Server itself is not shown, since its fault tolerance was not changed. It is handled by the Model 22x (Tandem System) as described in the TCOSS Configuration Manual [5].

Priority 2 is the higher priority. The line server will not accept a priority 1 connection if it is already connected with priority 2. On the other hand, it will disconnect from an already existing priority 1 connection if a connection with priority 2 is attempted. If a connection is closed, all active calls will be disconnected before.

If e.g. the first server becomes unavailable, the instance ECP on the second server will connect to the first line server.

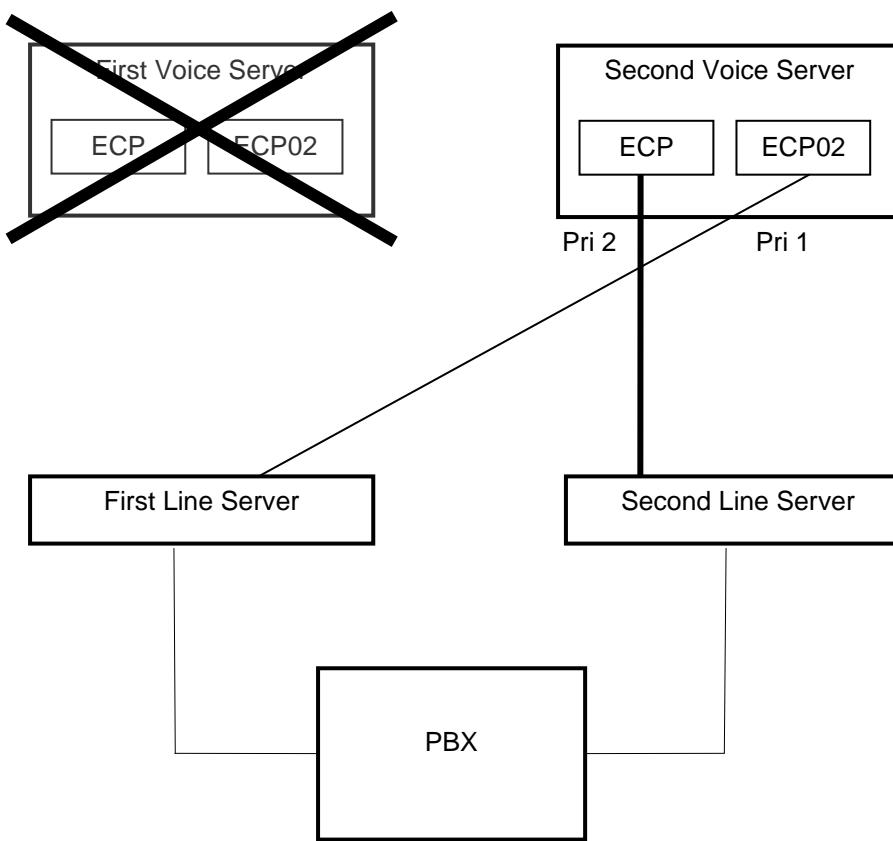


Figure 10 – Primary Voice Server Became Unavailable (i.e. Power Off)

As soon as the primary server is available again, its instance ECP will attempt to connect to the first line server and the instance of ECP on the second server will be disconnected again. The system will return to the state shown in Figure 9. The same scenario can be applied with the second voice server becoming unavailable.

4.9.1.3 **FoIPv3**

For fault tolerance via FoIPv3, refer to [\[23\]](#), [\[24\]](#).

4.9.2 **Voice Resource**

The VS offers an easy way to guarantee load balancing and fault tolerance within a VSG (see [Voice Server Groups and Remote Connections](#) for some background information).

TCRT is now able to automatically distribute calls to VS with the same resources (Figure 11 – Load Balancing Based on Resources) and to switch to a different VS with the same resource if the currently used is down (Figure 12 – Fault Tolerance Based on Resources).

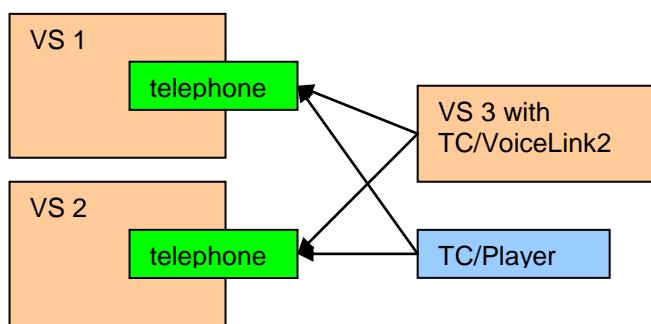


Figure 11 – Load Balancing Based on Resources

Note: The VS 3 does not offer the resource telephone. So the TC/Player will never try to connect to this VS.

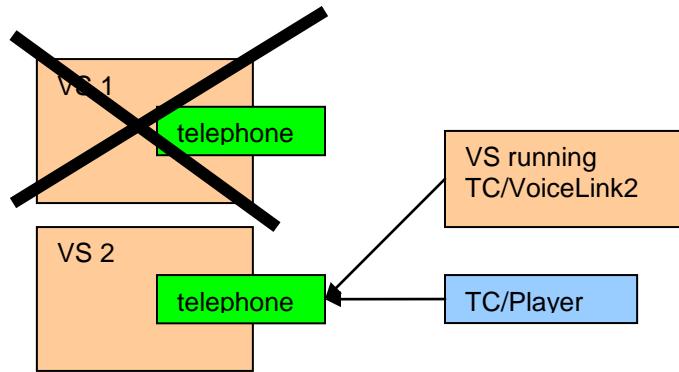


Figure 12 – Fault Tolerance Based on Resources

This is not only possible for outbound calls but also for inbound calls – primarily because TCRT does not distinguish between these two ways. (Figure 13 – Inbound call distribution)

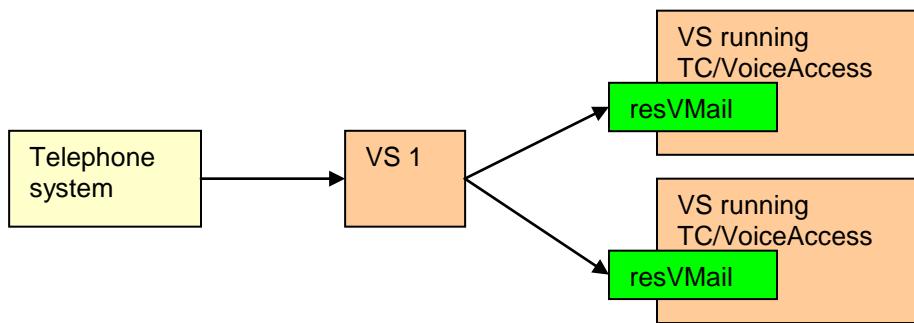


Figure 13 – Inbound call distribution

This is possible because the resources a VS publishes are not fixed but can be configured and with the TRS, even the assignment of a resource to a plug-in is possible.

4.9.3 Plug-ins

Like with the telephone system the fault tolerance and load balancing issues depend on the plug-in in use.

4.9.3.1 TC/VoiceAccess

VA only supports fault tolerance in terms of a TCOSS tandem system. Fault tolerance for SPIs must be supported by the SPI itself.

Load balancing for incoming calls has to be done with TCRT or the telephone system.

4.9.3.2 TC/VoiceLink2

Is the same as for VA. Fault tolerance has to be implemented in terms of a TCOSS tandem system.

For load balancing of outbound calls, the TCRT or a telephone system mechanism must be used.

The VL2 is not capable of load balancing the access to the TCOSS system.

Please note that the VL2 does not recognize other VS running the VL2. If both VS poll the same TCOSS they will do that regardless of the other VS. In terms of fault tolerance this means that it is not possible to leave one VL2 on hold. Both VL2 run all the time.

4.9.3.3 TC/VoiceXML

Load balancing and fault tolerance are not implemented by TC/VoiceXML. To access a web server, a hardware or a 3rd party software load balance can be considered.

For load balancing of inbound calls, the TCRT or a telephone system mechanism must be used.

4.10 Speech Profiles

The VS allows the parallel operation of multiple TTS engines with multiple TTS speakers. It is even possible to switch the TTS speaker and the TTS engine during playback dynamically.

The VS uses so called speech profiles to define which TTS engine and speaker should be used and when.

A speech profile is an exact definition of a TTS speaker. It defines a speaker with a specific speed and pitch. A speech profile has the following properties.

- A speech profile name
- An associated key
- A TTS speaker
- A speed
- A pitch

Each user can have up to 4 different speech profiles. Additionally, the SysUser can have up to 4 speech profiles. The user can access these speech profiles in VA where the keys 1-4 are the speech profiles of the SysUser and the keys 5-8 are the speech profiles of the user himself.

The user can configure his speech profiles with the TC Management Console [TC/MC] (see Figure 14). The SysUser already has 2 speech profiles (named "English" and "German") per default. These 2 default speech profiles should be adapted as soon as possible.

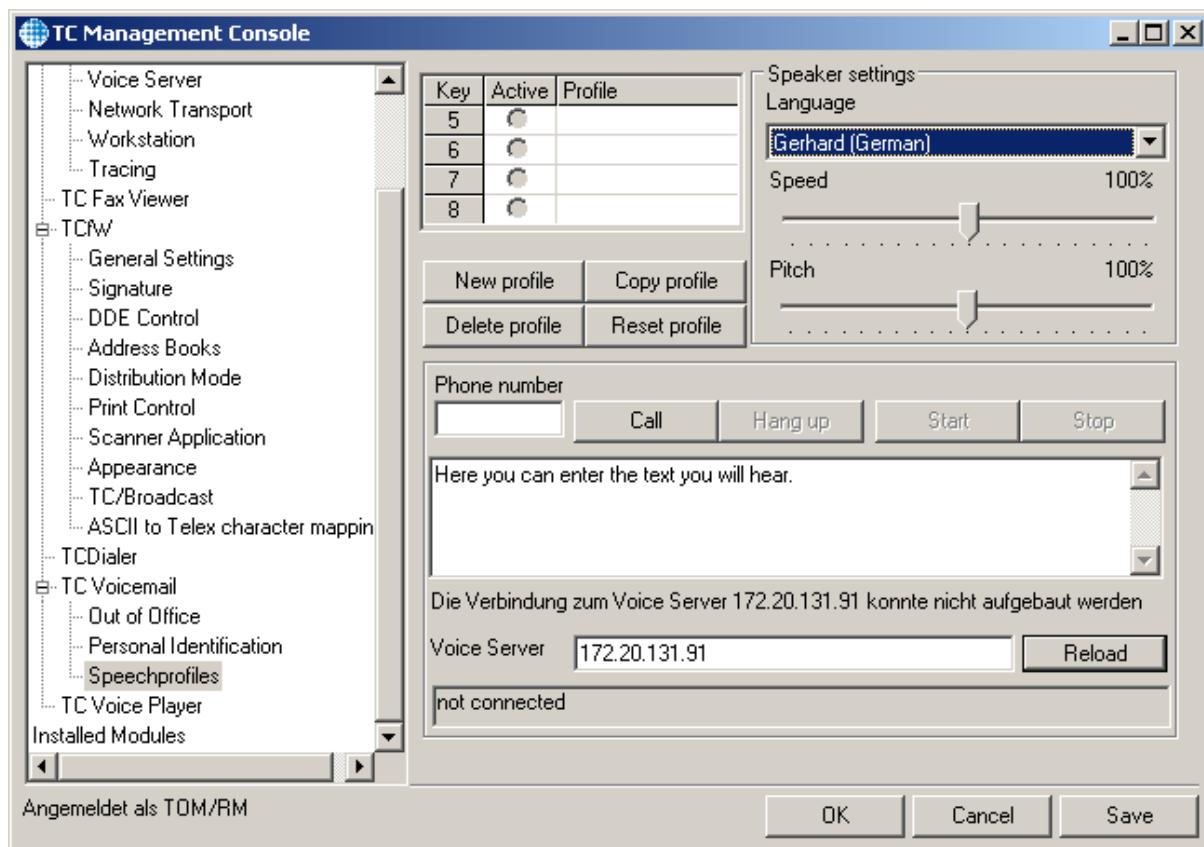


Figure 14 – TC/MC Speech Profiles

4.11 Service Provider Interface – SPI

The SPI is an abstraction layer between a mail client and a mail system, provided by TFC . This is used by VA to access the different mail systems.

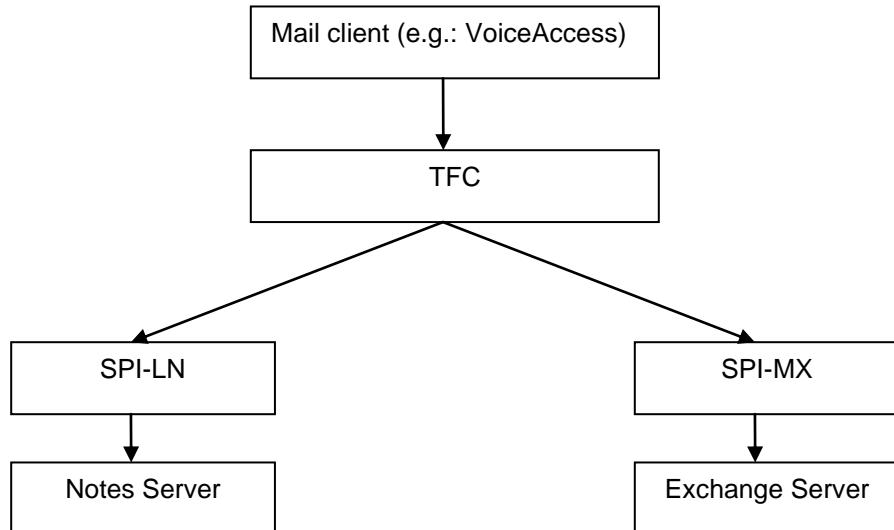


Figure 15 – SPI Architecture

This design provides great benefits when it comes to maintenance because there is only one implementation how to access a mail system and not four different ones.

During the VA setup, the user can choose which SPIs to install (see Figure 29 – VoiceAccess Basic Parameters 1).

4.12 TC/VoiceAccess

VA is an ActiveX-DLL supporting the COM1-Interface of TCECP:

It is a voice mail solution with an integrated attendant.

VA can interact with different mail systems simultaneously without having the user noticing any different behavior.

Messages can be sent from and to any mail system without any special addressing or routing information.

There are 2 basic call flows: Leaving a message and accessing the mailbox.

When a message is left, the VA receives a call, records the message and stores it in TCOSS. Regardless if the receiver of the message belongs to any other mail system, the message will always be stored in TCOSS and forwarded via TC/Link to the appropriate mail system.

When accessing a mailbox, VA checks the user status and option on TCOSS and receives the messages directly from the mail system the user belongs to.

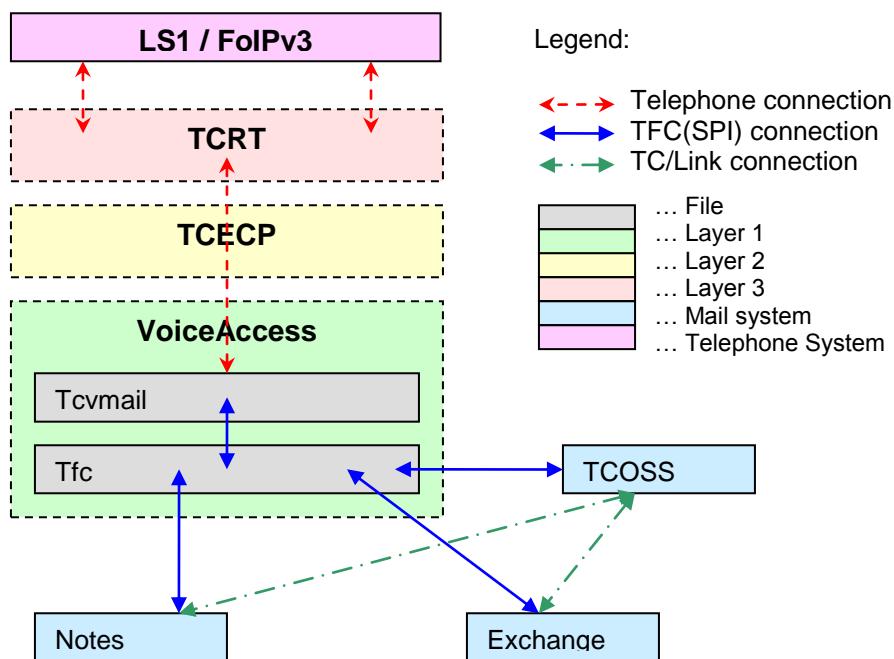


Figure 16 – VA General Architecture

Figure 16 shows a representation of the possible connections in a VA scenario.

To allow easy extensibility and customizability the workflow is not hard coded into the tcvmail.dll but uses a state machine with a state table. Every time before VA performs an action, the state machine checks what to do next and calls this function in VA.

A typical scenario can be seen in Figure 17. A connection is passed to VA (1). VA passes the control to the state machine (2). The state machine performs a lookup in the state table which action has to be done next (3). The state machine then calls this function (4) and VA executes it (5).

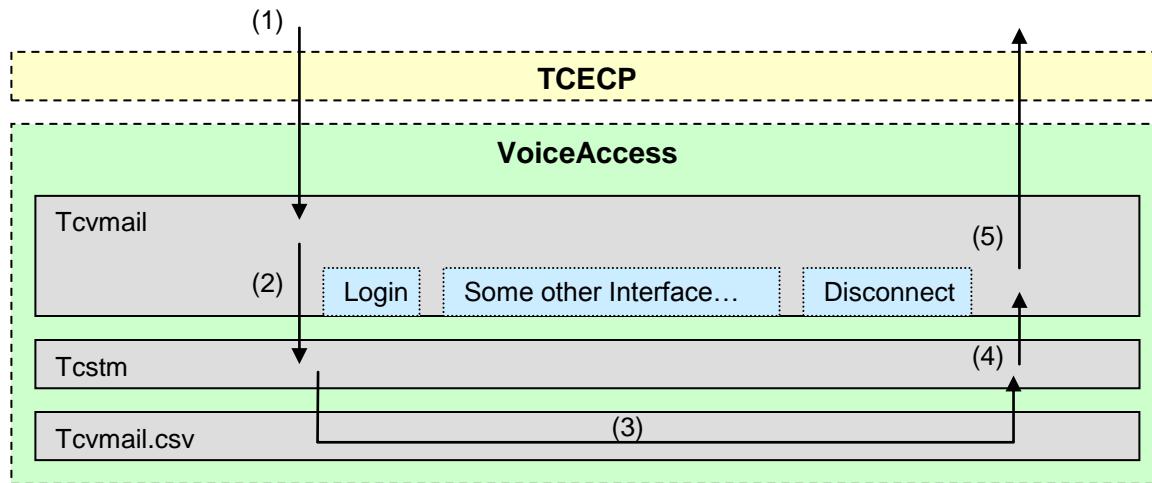


Figure 17 – VA – Typical Call Flow

4.12.1 Message Preprocessing and MetaMail Support (TC/PreProcess)

When playing a text message on the phone (using [Email Reader](#)), it is important that the text read by the TTS can easily be understood. Common abbreviations need to be resolved to be better readable. For instance, something like “:-)” has to be replaced by “smiley” or “ha ha”. Some TTS engines have so-called preprocessor modules to format text that way. Since they can be very different and their behavior is not always configurable, it is necessary to have an independent preprocessor.

With TC *MetaMail* it is possible to forward large messages to a mailbox of a third party mail system like Lotus Notes or Microsoft Exchange without sending the whole message. Instead the message on the third party system contains a so called *MetaMail URL* which is linked to the original message remaining on the Kofax Communication Server. This is what happens on the TC/Link server. On the voice server, the reverse operation must be done. If a user plays the messages in the inbox via telephone, the *MetaMail URL* will be resolved and the original message stored on the Kofax Communication Server will be played (or read if [Email Reader](#) is used).

Both, the message pre-processing and the *MetaMail URL* resolving, happen in a module called *TC/PreProcess*. It is called by TC/VoiceAccess every time when playing a message in the inbox.

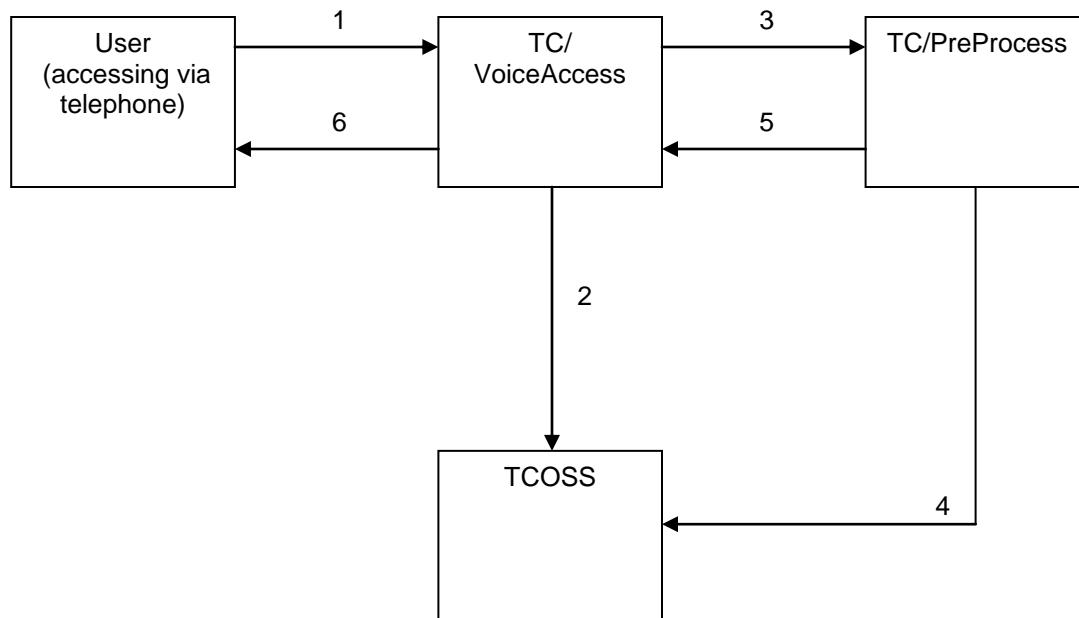


Figure 18 – Pre-Processor Workflow

1. User calls TC/VoiceAccess and logs on.
2. TC/VoiceAccess gets message from TCOSS
3. TC/VoiceAccess forwards a copy of the message to TC/PreProcess
4. Optional: TC/PreProcess accesses TCOSS to resolve MetaMail URLs
5. TC/PreProcess gives TC/VoiceAccess back a processed copy of the message
6. The message is read to the user listening on the telephone

4.13 TC/VoiceXML

The idea behind VoiceXML is a client-server approach very similar to a web-browser and a web-server. The VoiceXML client aka VoiceXML interpreter requests VoiceXML files from a web-server and “displays” them via phone to the user.

In the case of the TC/VoiceXML the interpreter is split into two parts. The first part is called the VXI-Client (tcvxclient.dll) and is the hardware abstraction for the interpreter itself (VXI.dll). In cases where a distinction between these two parts has to be done, they are referred as tcvxclient.dll and vxi.dll. Whereas TCECP communicates with the tcvxclient.dll, the tcvxclient.dll communicates with the VXI.dll over four additional interfaces. These interfaces represent different aspects of VoiceXML and the hardware abstraction. Namely:

- tcvxirec.dll for recording and recognition.
- Tcvxiprompt.dll for playback of prompts
- Tcvxitel.dll to control the telephone system (disconnect, transfer call ...)

- Tcvxiobject.dll for the <object> tag of VoiceXML

The VXI.dll uses a bunch of DLLs and 3rd Party Open Source Software (e.g.: OpenSSL, XercesC...) to retrieve and interpret the VoiceXML file. See Figure 19 for a simple illustration of the TC/VoiceXML architecture.

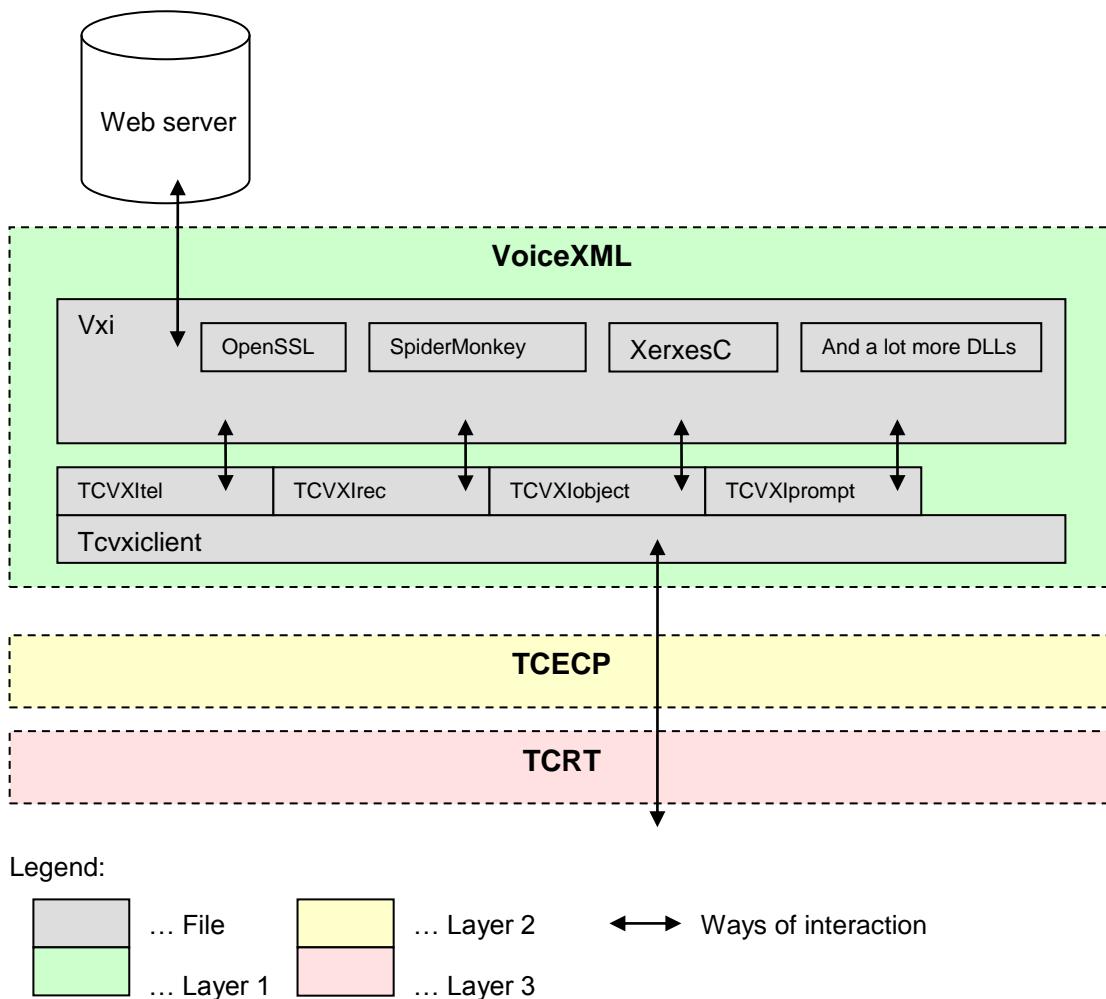


Figure 19 – VoiceXML Architecture

4.14 TC/VoiceLink2

The term TC/VoiceLink2 [VL2] describes a complete solution for sending voice messages. Unlike its predecessor TC/VoiceLink, the VL2 consists of multiple sub modules which allow easy extensibility.

The VL2 is an active voice system. It is more or less the counterpart of VA. With the VL2 it is possible to send messages to a telephone and get response. These responses may be: message is accepted, message is rejected, or the line is busy.

The advantages of the VL2 solution over the “old” VL are manifold.

While the VL was only capable to play two different menus, the VL2 can play different dialogs for every message. This is possible because VL2 uses VoiceXML for the playback.

The VL could only return very limited information about the actual call. The VL2 can handle every possible response and even generate reports out of it. This was made possible by introducing an object plug-in for VoiceXML.

The VL2 does accept various input types, ranging from plain text over VoiceXML to so-called user properties.

The typical installation scenario is shown in Figure 20.

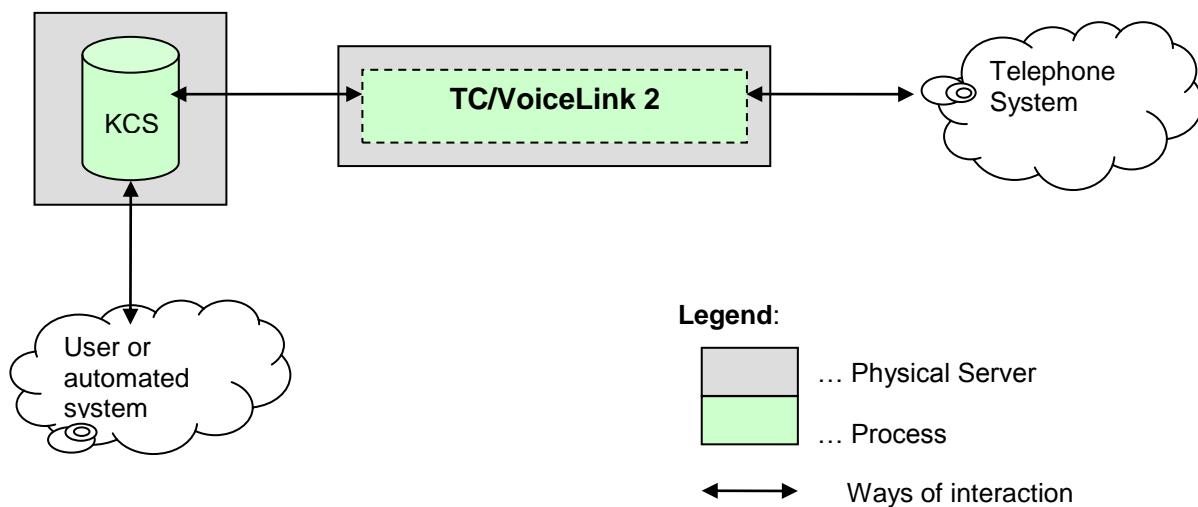


Figure 20 – VL2 Architecture

The scenario described an installation where KCS and the VL2 are installed on separate servers.

This guarantees that the KCS process does not interfere with the VL2 process. Otherwise, the VL2 could not perform a continuous playback only because KCS needs to OCR a bunch of incoming faxes.

The advantages are:

- Performance of one process is not influenced by another process
- Servers can be maintained separately
- Easily extendable

A typical workflow would look like this:

1. A user sends a voice message to the VL queue.
2. The VL2 retrieves the message from the queue.
3. The VL2 transforms the message to a VoiceXML file.
4. The VL2 calls the recipient and plays the VoiceXML file.
5. When the call is done, the VL2 generates a notification containing the response of the recipient.

As already mentioned, the VL2 is highly modular. Each key aspect of the VL2 has been built into modules to allow easy extensibility and customization. As seen in Figure 21, there are interfaces for logging & billing, message retrieval, and message playback. The VL2 core takes care of the call establishment and manages the different modules.

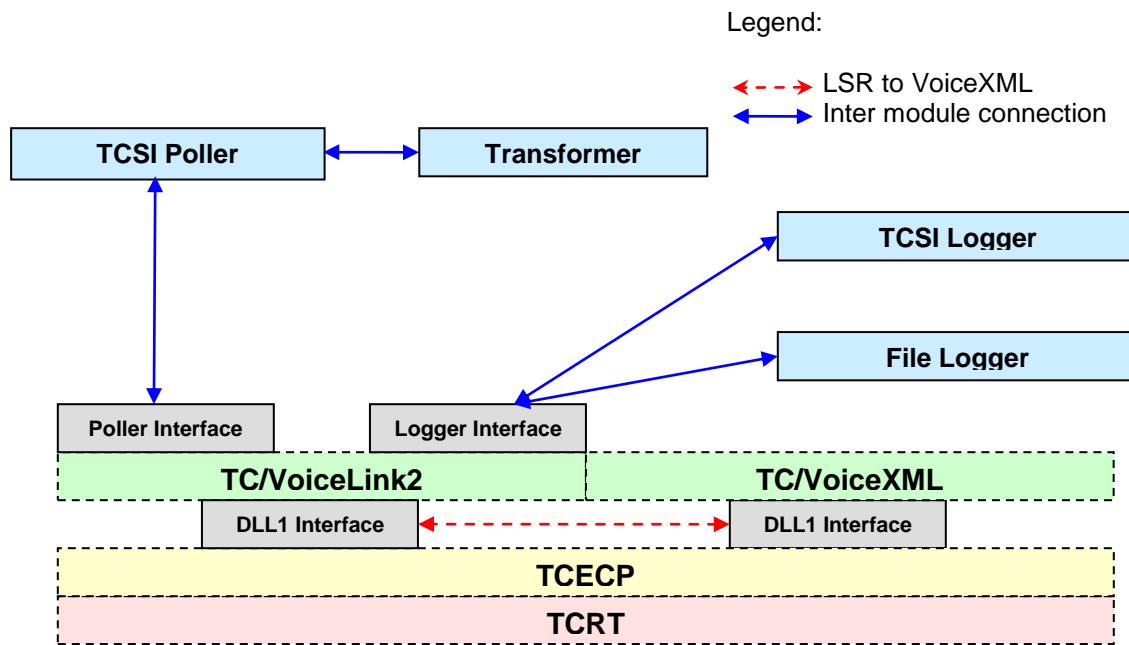


Figure 21 – VL2 Architecture and Modules

The VL2 retrieves messages from TCOSS via the TSCI poller .The VL2 can use unlimited pollers at the same time. For example, it is no problem to have two TSCI pollers running under one VL2.

The poller takes care that the message received from TCOSS is transformed into a format the VS understands. In this case it is VoiceXML.

This job is done by the TC/VLAtransformer which does the following:

- Checks if the message contains playable content
- Depending on the content, the message is not touched, extracted or transformed. If necessary the message will be transformed into a VoiceXML file
- Takes care of different languages
- Chooses the right template.

At the end, when everything worked, the result is stored as VoiceXML files into the “out” directory.

After retrieving and transforming the message, the VL2 establishes the connection to the destination number. If the connection was established successfully, the VL2 tells the VS to perform a LSR to the plug-in with the alias VoiceXML. This plug-in can now do the interaction with the recipient. When using TC/VoiceXML, the whole power of VoiceXML is ready to use.

If the recipient quits or the dialog was finished, the control of the session returns to the VL2. In the end, the VL2 tells the result of the call to the poller which will then close the message.

During the whole process, the VL2 logs events which can be used for logging and billing purposes. For the first time, it is easily possible to determine how many messages were successfully delivered over a certain period of time. Two different implementations are available. First logs directly to a file, whereas the second implementation uses the TCOSS logging facility. Unlike pollers, only one logger is supported at a time.

The design of the VS allows two basic scenarios for the VL2 to be used in.

The first and the most common scenario is that the VL2 runs on the VS with connection to the telephone system (see Figure 22).

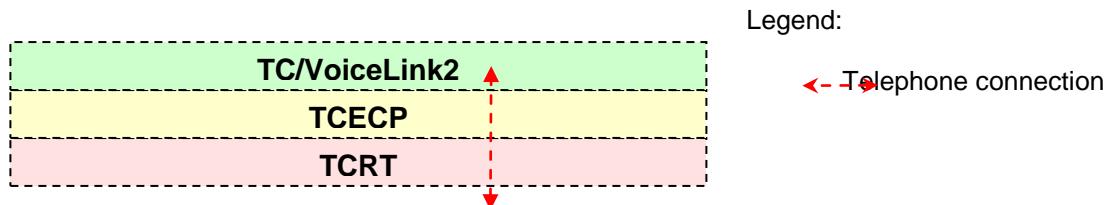


Figure 22 – VL2 on a VS with Telephone System Integration

In the second possible scenario, the VL2 runs on a dedicated VS. The VL2 will then establish a connection to second VS which is connected to the telephone system (see Figure 23).

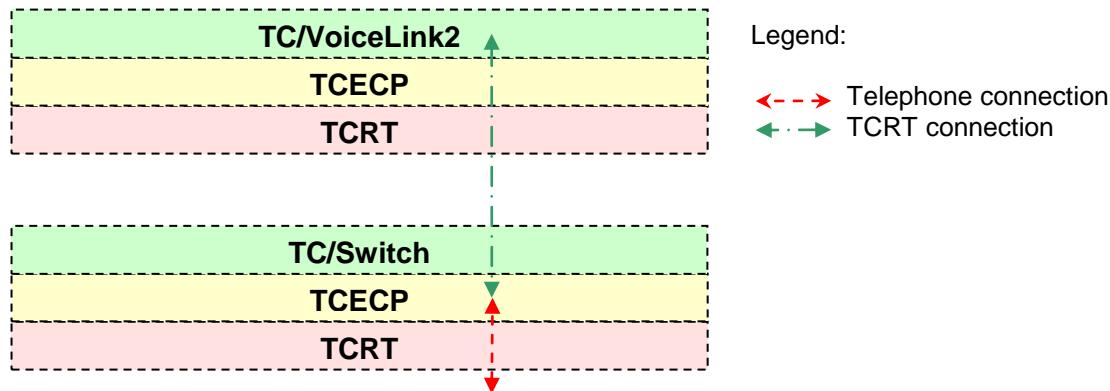


Figure 23 – VL2 Connecting to a Secondary VS

The scenario requires an additional computer; however, it offers several advantages:

- Fewer loads on the VS that is connected to the telephone system.
- It is possible that multiple of these VL2 servers connect to single VS: better scalability.
- Especially in high availability scenarios this configuration allows the separation of concerns.

The additional TC/Switch module is used to establish outgoing calls. TC/Switch is a script which is used for outgoing TC/Player connections. The VL2 takes advantage of this ability.

4.14.1 Polling

The way the polling works in the VL2 is quite different to the mechanisms in the VL. The polling is now more efficient and causes fewer loads on TCOSS.

The basic principle is that for every poller one worker thread is created. This thread is responsible for getting messages from TCOSS or a different message store.

E.g.: If the VL2 has 30 channels and only one poller configured, only one thread will be polling TCOSS instead of formerly used 30.

The errors influence the polling behavior in another way, too. Now, if an error occurs, the thread related to the channel which caused the error will be suspended. It is suspended until either a timeout occurs or another channel reactivates the poller. The reason for such a reactivation is most likely that the response to a message must be processed by the poller.

The following "Figure 24 – VL2 Polling Concept" shows a very basic scheme how the polling thread works.

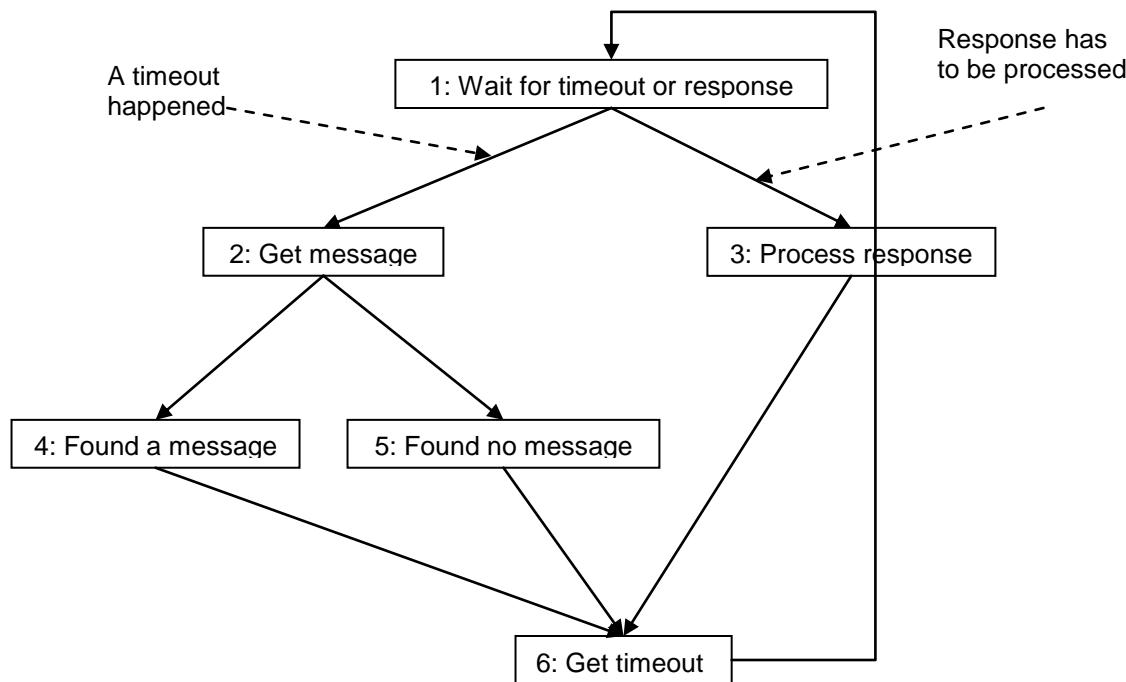


Figure 24 – VL2 Polling Concept

4.14.2 Response Handling

One of the new key features of the VL2 is the way how response of the recipient can be handled. While the traditional VL had very limited possibilities to tell the originator what was the response of the recipient like, the VL2 extends the existing one and implements two additional methods.

The method of the VL was to terminate the send order in a specific way and to add an explanatory comment to it. How the send order was to terminate was determined by the state of the VL and by the actual error code. Both, state and error code, had a fixed range. This was sufficient because the VL had limited interaction capabilities. The VL2 extends this by removing the state and by adding the possibility to check against a so-called response string. This response string is a free text which is set during the IVR dialog. This offers virtually unlimited possible error <-> response combinations.

The second method uses the response string provided by the IVR dialog to create a notification to the originator. The notification contains this response string. Because the response string is not limited in size a notification may contain anything from a simple “ACCEPTED” to a complete purchasing order.

The third method is a direct result of the usage of VoiceXML for the IVR dialog. By using VoiceXML, it is possible to submit requests and responses in real-time to a web server or to a back end system where it is processed.

To set a response string, VoiceXML offers three new object methods.

- com.topcall.SetResponse ... Sets the response to the specified value.
- com.topcall.AppendResponse ... Appends the specified value to the response string.
- com.topcall.ResetReponse ... Clears the response string.

4.14.3 Templates

For the transformation of the message, so-called templates are used. In these templates placeholders are replaced by the content of the message or specified TSCI properties.

A template could look like this (without the xml-header):

```
<form id="formPlayMsg">
    <block name="msg">
        <prompt>
            You got following message:
            $body.text$
        </prompt>
    </block>
</form>
```

where \$body.text\$ is replaced by the content of the body text of the message.

Detailed information about how to write templates can be found in the chapter [TC/VoiceLink2 – The Transformation of a Message](#)

4.14.4 Multilingual

Another major feature of the VL2 is the support of multilingual jobs. If the VL2 detects that the recipients of a message require different languages, this is taken into account. This is done with different templates for the different languages. If no language is specified, the VL2 will look for a file which looks like “*templatename.templ*”. If a language is specified, the VL2 will insert the language identifier after the template name. The resulting file looks like this: “*templatename-languageid.templ*”. If no such file is found the file without the language identifier will be used. In the end, there must always be a valid VoiceXML file.

Example: A message has two recipients with the language identifiers “EN”, “DE”, and one recipient without language identifier. The message should use the template “template1”

In case there are the following two templates: “template1.templ” and “template1-DE.templ”, one template (for the language “EN”) would be missing. Nevertheless the VL2 does not know this and expects an “index-EN.vxml” file in the “out” directory. To achieve this, the VL2 uses “template1.templ” also for the language “EN”. As a result, the following three files will be created:

“index.vxml”, “index-DE.vxml” and “index-EN.vxml”

4.14.5 Types of Messages

The VL2 distinguishes four different message types.

Each type is treated differently. The four types are:

Type name	Possible Message Content	VL2 Behavior
URL	Only a file://, http:// or https:// URL	The VL2 will assume that a valid VoiceXML can be found at the location of the URL and will tell VoiceXML to use this URL for playback.
VOICEXML	A valid VoiceXML document	The VL2 does no template replacement but use this document for playback.
USERPROPERTIES	XML which is validated against the VL2 user properties DTD.	The VL2 uses the user properties for the template conversion.
TEXT	“plain” text. Text as it is written in emails and does not fit into one of the above categories. It must not start with “<?xml”	The full program. Transformation and conversion.

Table 4 – VL2 Message Types

4.14.6 User Properties

As mentioned in the chapter [Types of Messages](#), the VL2 knows something called “user properties”.

These user properties describe a way how a message can contain, besides the TSCI Properties, additional properties in multiple languages. The possibility to send one message to multiple recipients in multiple languages is the most important point of the user properties.

User properties are written in XML and are validated against the following DTD schema:

```
<!ELEMENT parameter (#PCDATA)>
<!ATTLIST parameter name CDATA #REQUIRED>
<!ATTLIST parameter value CDATA #REQUIRED>
<!ATTLIST parameter xml:lang CDATA #IMPLIED>
<!ELEMENT userproperties (#PCDATA | parameter)*>
```

This allows to write messages with the following body text:

```
<?xml version="1.0" ?>
<!DOCTYPE userproperties PUBLIC "http://www.topcall.com/voicelink2" "default">
<userproperties>
<parameter value="default Text" name="TEXT"/>
<parameter value="deutscher Text" xml:lang="DE" name="TEXT"/>
<parameter value="English text" xml:lang="EN" name="TEXT"/>
<parameter value="4711" name="AMOUNT"/>
</userproperties>
```

The above example does make the following properties available to the VL2.

- “TEXT” in the default language. This line is MANDATORY.
- “TEXT” in the language “DE” and the value “deutscher Text”
- “TEXT” in the language “EN” and the value “English text”
- “AMOUNT” in any language with the value “4711”

Note: It is mandatory that each multilingual parameter is also defined without xml:lang attribute.

The code-snip from “sample.templ” with the content:

```
<block>
<prompt>
The text is a $TEXT$ . You owe me $AMOUNT$
</prompt>
</block>
```

will be then transformed to:

```
<block>
<prompt>
The text is a default text . You owe me 4711
</prompt>
</block>
```

The code-snip from “sample-EN.templ” with the content:

```
<block>
<prompt>
The text is an $TEXT$ . You owe me $AMOUNT$
</prompt>
</block>
```

will be then transformed to:

```
<block>
<prompt>
The text is an English text . You owe me 4711
</prompt>
</block>
```

whereas the code-snip from “sample-DE.templ”:

```
<block>
<prompt>
Dieser Text ist ein $TEXT$ . Du schuldest mir $AMOUNT$
</prompt>
</block>
```

will be then transformed to:

```
<block>
<prompt>
Dieser Text ist ein deutscher Text . Du schuldest mir 4711
</prompt>
</block>
```

To enable the VL2 to recognize a text block as user properties the text must start with:

```
<?xml version="1.0" ?>
<!DOCTYPE userproperties PUBLIC "http://www.topcall.com/voicelink2" "default">
```

The user properties must conform to the DTD.

Find additional examples in the chapter [TC/VoiceLink2 – The Transformation of a Message](#)

4.14.7 Text2Wave Pre-Conversion

During the transformation of the message to VoiceXML, the VL2 can also convert the text of the message into a wave file before the actual call starts. This feature is actually part of the TC/VLATransformer and therefore only available if the TSCI poller is used. The TC/VLATransformer does not fully interpret the VoiceXML file but only looks for <prompt> tags. Only <prompt> tags without any dynamic content will be converted and replaced.

E.g.: The following is part of a VoiceXML file:

```
<block>
<prompt> Hello World </prompt>
</block>
<block> This is a test </block>
```

In the above case the result could look like this:

```
<block><prompt><audio src=".//ac123.wav">Hello World </audio></prompt></block>
<block>This is a test</block>
```

Even if the VL2 cannot retrieve the wav file, nothing will be lost because the original text is still present as alternative text.

This feature must be used with care. The conversion is done on the same server as the call is done, so this has a big influence on the performance of the VS. Furthermore the conversion is inefficient because it converts every possible text, even if the text is never played at all.

But it also has advantages. If the TTS engine is slow, which means that the called person would hear nothing for several seconds before a prompt is played, the pre-conversion can reduce or even eliminate this problem.

4.15 VoiceXSP

The Voice xSP solution follows the concept of TC Storage/Media Server:

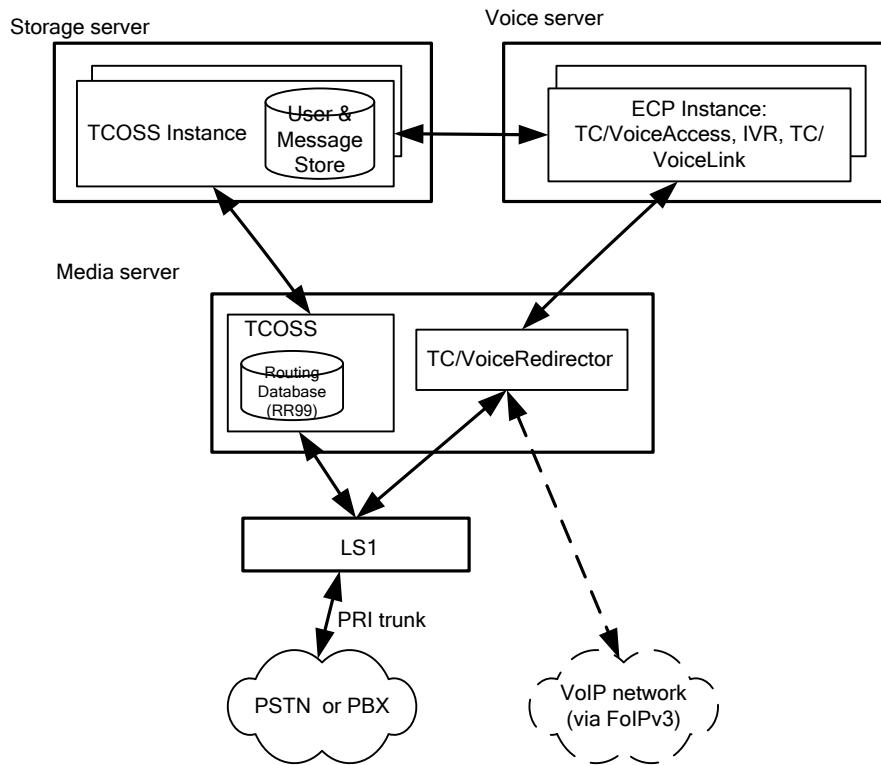


Figure 25 – VoiceXSP Concept

- Customer specific data are stored on the Storage server where one TCOSS instance per customer is running.
- On the Voice server one TCECP instance per customer is running. The advantage is that different customers may have different settings like prompts, languages, TTS engines etc.
- There are one or more media and line servers being shared by all TCOSS and TCECP instances on the Storage server. On each media server one routing TCECP instance is running (TC/VRedirect), one per connected LS1 server and it is connected with the LS1 server through the LS1 engine. The role of TC/VRedirect is to route incoming call towards multiple TCECP instances based on routing information (RR99) provided by TCOSS
- Voice messages are stored in the corresponding TCOSS storage instance and forwarded to the customer's collaboration system (mail server) by TC links.

For more details about the Voice xSP solution, refer to [\[10\]](#).

5. Requirements

5.1 Recommended

It is very difficult to specify a minimum requirement for a VS. There are too many possible combinations of modules and scenarios how the VS is used. Therefore instead of specifying minimum requirements the document gives recommendations for different scenarios. These scenarios are explained in details in the chapter [Scenarios And Tutorials](#)

Optional a soundcard can help troubleshooting

Last but not least, a working and supported [telephone system](#) is mandatory.

Additional information about the performance can be found in the [Performance](#) chapter.

5.2 Additional Requirements for ISDN

Please see [\[3\]](#) for details on the additional requirements when using the ISDN (with line server) as telephony system.

5.3 Additional Requirements for H.323 / SIP

Please see [\[23\]](#) for details on the additional requirements when using H.323 or SIP (with FoIPv3) as telephony system.

5.4 Additional Requirements for TC/VoiceAccess

Depending on the situation and environment VA will need additional modules.

To support foreign mail systems, the corresponding SPIs are necessary. Where the SPIs are installed automatically they have certain requirements about the foreign mail system:

- Lotus Notes must be version 7 or higher.
- MS Exchange Server must be Exchange 2007, 2010, 2013 or 2016.

Additionally to the SPIs the corresponding TC/Link is necessary:

- TC/LP 2.07.00 or higher.

To use the [email reader](#) functionality of VA, following components are necessary:

- Microsoft SAPI 5
- A TTS Engine (see [Supported TTS Engines](#) for supported TTS engines)

5.4.1 Recommended Number of Voice Channels

The following table is only a recommendation for the number voice channels required for an installation. Please note that these numbers might vary depending on the real usage scenario.

No. of voice mail users	25	85	250	425	625	2300	6000	10000
Voice channels (lines)	2	4	8	12	16	48	100	120

Factor	12	20	30	35	40	50	60	85
--------	----	----	----	----	----	----	----	----

Table 5 - Number of Voice Channels

5.5 Additional Requirements for TC/PreProcess

- TC/LP 2.05.08 or higher if MetaMail without URL encryption is used,
TC/LP 2.09.03 or higher if MetaMail with URL encryption is used.
(TC/PreProcess can only resolve *MetaMail URLs* generated by TCAt2URL from TC/LP)
- Sufficient TFC licenses installed on the TCOSS and archive servers if *MetaMail* is used

5.6 Additional Requirements for VL2

- Kofax Communication Server version 7.80.04 or higher
- To use the answering machine detection feature in a remote VS scenario, the remote VS KCS must be at least version 7.66.01.
- A [supported TTS](#) engine, capable of SAPI 5, must be installed on the VL2.
This is necessary, because the VL2 is not shipped with any pre-recorded prompts.
- **Note:** It is possible to run the VL2 without a TTS engine but VoiceXML is required to use only pre-recorded prompts.
- The VL2 supports Windows Server 2008, and 2012.

5.6.1 Concurrent Voice Channels

Estimate the demand of concurrent voice channels according to the formula below:

time_greeting = the time that is used to play a greeting. With the standard prompt it will take 6 seconds.

time_dialogs = the time the dialogs are using. The first dialog needs 5 seconds. The second dialog needs 7 seconds.

time_body = the time the real message will use.

min_message_time = the minimum amount of time on single message will need.

reserve_time = for the case that some recipients want to hear the message again or do not pick up the receiver immediately. The *reserve_time* should be ~1/3 *time_body*.

message_time = how long a message in average will need to be played.

message_count = the number of messages the VL should send.

max_time_to_use = the maximum time the VL is allowed to use for the specified count of messages

min_lines = the minimum count of lines the VL will need to send all messages in time.

min_message_time[sec] = *time_greeting[sec]* + *time_dialogs[sec]* + *time_body[sec]*

message_time[sec] = *min_message_time[sec]* + *reserve_time[sec]*

min_lines = $\frac{\text{message_time[sec]} \times \text{message_count}}{\text{max_time_to_use[min]}}$

For example:

$time_greeting=6 \text{ sec}$
 $time_dialogs = 12 \text{ sec} (\text{both dialogs are on})$
 $time_body = 23 \text{ sec}$
 $time_reserve = 9 \text{ sec}$
 $message_count = 1000$
 $max_time_to_use = 60 \text{ min}$
 $min_message = 6 + 12 + 23 = 41 \text{ sec}$
 $message_time = 41 \text{ sec} + 9 \text{ sec} = 50 \text{ sec}$
 $min_lines = \frac{(50 * 1000) / 60}{60} = 13,8 \approx 14$

At least, 14 lines are necessary to send 1000 messages within an hour.

On the TCOSS server, at least one C0 channel is required per TSCI poller.

5.7 Additional Requirements for TC/VoiceXML

SAPI 5 and a [supported TTS](#) engine

5.8 Additional Requirements for TC/SPI-LN

A Lotus Notes client must be installed on the workstation.

An ID-File is needed. If the ID-File has a password, it is recommended to install the Notes add-in "TCPWD.dll". Set the registry key "... \NotesAddin\TCPWD" [Reg_SZ] with the password of the ID-file, so the SPI is allowed to login automatically.

The "TCPWD.dll" will be installed with the KCS.

This user must have read and write rights to all users' mailboxes (manager rights which are needed to get the correct read/unread list of the user of the mail database), so the administrator has to add this user to the ACL of all user mailboxes which should be supported.

Required TC/Link-LN dirsync settings:

Depending to the TC/Link-LN (TCUserName) username settings, the inactive Notes address of the user must be in the person document of the notes address book.

It is required for the SPI to find the correct user's mailbox.

Usage in Domino Cluster: all user mailboxes should have the setting: "Replicate unread marks" (see on the last tab on the Database Properties dialog box) = "Clustered servers only" or "All servers" – for unread mark replication.

5.9 Additional Requirements for TC/SPI-MX

TCLINK:

Link type TC/LINK-MX7 must be used to interconnect Exchange and TCOSS. TC/SPI-MX does not support TC/LINK-SM.

Install TC/LINK-MX7 before installing VoiceMail; KCS setup asks for a configuration file created by the link.

The link and TCSPIMX must use the same Exchange message class for voice messages.

Required TC/LINK dirsync settings:

USR\O\ClientSettings=1, USR\O\KeepFaxPassword=1.

Otherwise, dirsync deletes fax password and voice prompts.

Windows security:

TC/SPI-MX connects to its Exchange mailbox via MAPI/EWS. With Exchange version 2007, it connects directly to the Exchange mailbox server where the mailbox is located. With Exchange 2010 or 2013, MAPI connects to an Exchange client access server. For Exchange 2013 or 2016, EWS connects to an Exchange mailbox server where the mailbox is located. In this manual, the term “Exchange Server” usually refers to this server.

The computer where TC/SPI-MX runs must be either in the same domain as the Exchange server, or in a domain that has a trust relationship with the Exchange server’s domain.

The computer must not be part of a Windows cluster.

Connection to Exchange:

TCSPIMX can only access Exchange mailboxes via the network. Remote sites connected via dial-up connections cannot be reached.

In an IPv6 environment, the MAPI/EWS access to Exchange might not work. In this case, do one of the following:

- Use the IPv4 address in the registry value ExServer/ServerUrl.
- Add the name and the IPv4 address of the Exchange server to the hosts file.

Shadow users:

To access voice messages via telephone, the Exchange user needs a shadow user profile on TCOSS.

Each profile must contain an inactive address with the service “VOICE” (the user’s mailbox number). An additional inactive address holds the address of the Exchange mailbox.

TC/SPI-MX can only work with Exchange shadow users created by TC/LINK-MX7.

These shadow users have an inactive address with service “MX7” and the user’s primary SMTP address.

Configuration:

Setup sets the following registry values:

Registry Key	Value	Description
SPI\Exchange	MX7	Service for inactive Exchange address
SPI\Exchange\ExAddrTypeEX	SMTP	Exchange address type matching this service
SPI\Exchange\TCQueue	TCLMX7Q	Base queue name used by the link

5.10 Required LAN Bandwidth

As the required LAN bandwidth depends on many factors, merely a recommendation can be given.

The typical audio stream is 8kHz 8bit mono. This sums up to uncompressed 64kb/s (Kilo bit per second). This would be the minimum bandwidth one call needs. But there is some overhead in the communication and it is also wise to keep some buffer. In the end, 100kb/s or 10kB/s (Kilo byte per second) is the required bandwidth for one call.

As already hinted, this counts only for an uncompressed audio stream. Audio codecs like G729a require less bandwidth because of their compression.

This 10kB/s are required for a call to perform playback or recording unidirectional. It is sufficient to calculate the unidirectional case as long as ASR is not involved. As soon as ASR is used, this doubles because ASR requires some kind of recording even during playback.

If a call has to be forwarded, e.g. if a user plays a wav file with the TC/Player over telephone, a call will be established from the user's PC to the VS. This call will then be forwarded to the telephone of the user. Now we have two calls, each of them using 10kB/s.

The next factor is that the plug-ins typically store or retrieve data from the network. VA stores the voice mails on the TCOSS server in a continuous stream with ~10kB/s (if the audio format is uncompressed). The VL2 polls the TCOSS for new message and loads the audio and VoiceXML files from a file/web server. It is not possible to say how many kB/s such an action needs. This action does not require streaming and is therefore processed as fast as possible. This causes short but significant peaks in the network load. As already mentioned, it is not possible to give exact numbers about how many kB/s should be reserved per VL2 instance. The audio and VoiceXML data will be cached on the VS to reduce network load further.

At the end we get the following numbers:

- For each simultaneous call to or from the telephone system, the VS needs at least 10kB/s.
- For each simultaneous call to VA, additional 10kB/s are required, regardless if the caller leaves or listens to a message.
- For each simultaneous connection from a TC/Player to the VS, 10kB/s must be calculated.
- For each VL2 instance an average peak of 100kB/s should be planned. Although this bandwidth is not required all the time, this bandwidth should always remain free.

It is recommended to use a dedicated LAN-connection between the VS and the LS1 server. This keeps the connection unaffected of peaks in the network load between VS and TCOSS.

6. Licensing

In the price list there are several products and components listed which are necessary to activate a TC Voice Service.

First of all, the 'Server operating system for voice communication' license is required. This includes the basic VoiceAccess functionality and two voice channels. For additional channels, two items on the price list are available:

- 'Voice channel' for traditional PABXs, in this case, please note that also hardware (line server) is needed.
- 'Voice over IP channel' for IP-based PABXs

In addition, optional components for VoiceAccess can be activated - they are separately licensed:

- Email-Reader (See section [Email Reader Support](#))
Note: The Email-Reader license does not include the TTS engine itself. Therefore an own pricelist is available or please contact a vendor of your choice.
- TC/Attendant (See section [The Attendant](#))
- TC/PreProcess (See [additional requirements for TC/PreProcess](#))
- TC/VoiceXML (See section [TC/VoiceXML](#))

In addition to the VS, VL2 can be activated with an own license which is required for outbound scenarios. Because the VL2 requires VoiceXML also a TC/VoiceXML license must be available. When purchasing a VL2 license a TC/VoiceXML license is included for free.

The following optional client applications are available:

- TC/Player [17]
- TC/Web [16]

Note: If TC/MetaMail is used one of the above clients is required.

To integrate VoiceAccess into existing mail systems the following TC/LINK license is additionally needed:

- Microsoft Exchange environment requires TC/LINK-MX7
- Lotus Domino environment from IBM requires TC/LINK-LN

For more information about licensing TC/VoiceAccess please, ask your sales contact.

The VS, especially TC/VoiceXML, uses a lot of 3rd party components. Their license conditions can be found in the chapter [3rd Party Licenses](#)

7. General Module Information

7.1 General

This chapter holds information which do not fit into any other chapter.

7.2 TC/VoiceAccess

Interface: COM1

Type: Passive

Name: "TCVoiceMail.VoiceMail/TCVoiceMail.Application"

Filename: "tcvmail.dll"

Dynamic Reconfiguration: Not supported

7.3 TC/VoiceLink2

Interface: DLL1

Type: Active

Name: "C:\TCOSS\System\TCVLink2.dll"

Filename: "tcvlink2.dll"

Dynamic Reconfiguration: Essentially the whole configuration. Even new pollers are loaded dynamically.
The only thing which is not supported is unloading pollers.

7.3.1 TCSI Logger

Filename: "tcvl2logTCOSS.dll"

Dynamic Reconfiguration: Trace level, server, path, user name, password...

7.3.2 File Logger

Filename: "tcvl2logFile.dll"

Dynamic Reconfiguration: The Trace level and the log file name.

7.3.3 TCSI Poller

Filename: "tcvl2PollTCOSS.dll"

Dynamic Reconfiguration: Everything with one possible exception. If there are still messages open, the TCOSS server will not be changed.

7.4 TC/VoiceXML

Interface: DLL1

Type: Passive

Name: "C:\TCOSS\System\tcvxiclient.dll"

Filename: "tcvxiclient.dll"

Dynamic Reconfiguration: The name of the configuration file is reloaded. The trace settings and the name and location of the base VoiceXML file are also reloaded. The list of plug-ins for the object interface is reloaded as well. New plug-ins are loaded if necessary, but not unloaded.

8. Installation

8.1 Where to Install a Voice Server

It is not recommended to install the Voice server on a TCOSS or Link Server due the following limitations:

- A VS is a real-time system, and needs all the resources it can get. Under heavy load it may happen that the VS cannot response to an incoming call because TCOSS uses all resources.
- Fail-safe: Maybe the complete (restarting Windows) Voice server has to be restarted. If the Voice server is located on the same server as TCOSS, TCOSS will also be restarted.

There are some additional limitations, for example: If using VA in a scenario where it should access Lotus Notes mailboxes. Please see the [restrictions](#) for TC/SPI-MX and TC/SPI-LN for more details.

The recommendation must be that a VS should be installed on a stand-alone box with as little as possible additional software.

8.2 What Should Not Be Installed on a Voice Server

The KCS Client Applications should not be installed on a voice server because of different DLL versions and different directories for the same version of DLL.

8.3 Installing Prerequisites

8.3.1 TC/SPI-LN

- Install the Notes Client on the voice server. Use the “single user” installation variant.
- Append the Notes Path (e.g.: C:\Notes) to the Environment variable “Path” (this is important to register the SPILN during setup!)
- Install the ID-File which must have MANAGER right for the mailboxes which should be accessed by SPILN.

8.3.2 TC/SPI-MX

The preliminary steps depend on the version of the Exchange organization:

2. If the link to Exchange does not run on the Voice Server, copy the TCARCHEX.INI file from the link computer to the Voice Server.
This file is created at TC/LINK-MX7 startup and can be found in directory C:\TCOSS\TCLP.
The link instance name is appended to the file name; so the default INI file created by TC/LINK-MX7 is called TCARCHEX-TCLINKMX7.INI.
3. The Voice Server must be member of the same forest as the Exchange Server, or of a domain that is trusted by the domain of the Exchange Server.
4. Create a domain user with an Exchange mailbox for TCSPIMX.
Add this user to the Domain Admins group.
5. To configure a user for accessing mail server:
 - For a user connecting Exchange Server using MAPI:
 - a) Grant this user full access to all mailbox stores:

- b) Use the Exchange Management Shell for granting full access to all mailbox stores. Use the following command:

```
get-MailboxDatabase -Server <server> | add-ADPermission -User <user> -AccessRights "GenericAll"
```

In the above example, <server> is a placeholder for the Exchange server name (simple server name, no FQDN), and <user> stands for the user name. For server "EX07" and user "TCSPIMX" the command would be:

```
get-MailboxDatabase -Server "EX07" | add-ADPermission -User "TCSPIMX" -AccessRights "GenericAll"
```

- c) Then restart the Exchange information store service, to make the permissions effective.
- o For a user connecting Exchange Server using EWS:
 - a) Configure impersonation for all users in an organization. Use the Exchange Management Shell to configure impersonation for all users
[https://msdn.microsoft.com/en-us/library/office/dn722376\(v=exchg.150\).aspx](https://msdn.microsoft.com/en-us/library/office/dn722376(v=exchg.150).aspx).
 - b) Use following command to add the impersonation permission to enable a specific user account to impersonate all other users.

```
New-ManagementRoleAssignment –name:impersonationAssignmentName –Role:ApplicationImpersonation –User
```

- c) The following example shows how to configure impersonation to enable a specific user account to impersonate all other users.

```
New-ManagementRoleAssignment –Name:SPIMX_EWS –Role:ApplicationImpersonation –User: "kcctest16\TCSPIMX1"
```

6. Add this user to the Administrators group of the Voice Server. If the Voice Server is a domain controller, this can be done via Administrative Tools | Active Directory Users and Computers. If it is a normal PC, use Administrative Tools | Computer Management | Local Users and Groups.
7. Additionally, check the right to "Logon as a batch job" on the Voice Server. This can be done via Administrative Tools | Local Security Policy | Local Policies | User Rights Assignment on normal PCs, and via Administrative Tools | Domain Controller Security Policy | Local Policies | User Rights Assignment on domain controllers.

TCSPIMX needs the Windows messaging components to communicate with the Exchange server. These components are not part of a standard Windows installation, nor a part of Outlook. Therefore, the following components must be installed on the Voice Server:

- ExchangeMapiCDO (standalone MAPI), downloadable from Microsoft Web Site.

The following table shows the minimum ExchangeMapiCDO versions to be used.

Exchange server version	ExchangeMapiCDO version	ExchangeMapiCDO date
Exchange 2010	6.5.8147.0	December 2009
Exchange 2013	6.5.8320.0	May 2013

Current download location:

<http://www.microsoft.com/en-us/download/details.aspx?id=39045>

8.3.2.1 Check Exchange 2010 Throttling Policies

Exchange 2010 by default limits the resources that each user consumes. By default, a user can only have 20 concurrent connections to a server. To bypass the default throttling policy, you can define a policy that applies only to the domain user for TCSPIMX.

The following example shows how to assign a new throttling policy called “SPIMXPolicy” to user “TCSPIMX”. The new policy explicitly disables throttling mechanisms for MAPI connections.

Open the Microsoft Exchange Management Shell and type:

- New-ThrottlingPolicy SPIMXPolicy
- Set-ThrottlingPolicy SPIMXPolicy -RCAMaxConcurrency \$null -RCAPercentTimeInAD \$null -RCAPercentTimeInCAS \$null -RCAPercentTimeInMailboxRPC \$null
- Set-Mailbox “TCSPIMX” -ThrottlingPolicy SPIMXPolicy

Note: It can take up to 20 minutes until the new throttling policy is used by all servers.

8.3.2.2 Check Exchange 2013 Throttling Policies

Exchange 2013 by default limits the resources that each user consumes. By default, a user can only have 40 concurrent connections to a server. To bypass the default throttling policy, you can define a policy that applies only to the domain user for TCSPIMX.

The following example shows how to assign a new throttling policy called “SPIMXPolicy” to user “TCSPIMX”. The new policy explicitly disables throttling mechanisms for MAPI connections.

Open the Microsoft Exchange Management Shell and type:

- New-ThrottlingPolicy SPIMXPolicy
- Set-ThrottlingPolicy SPIMXPolicy -RCAMaxConcurrency Unlimited
- Set-Mailbox “TCSPIMX” -ThrottlingPolicy SPIMXPolicy

Note: It can take up to 20 minutes until the new throttling policy is used by all servers.

8.3.2.3 Check Exchange 2016 Throttling Policies

Exchange 2016 by default limits the resources that each user consumes. By default, a user can only have 40 concurrent connections to a server. To bypass the default throttling policy, you can define a policy that applies only to the domain user for TCSPIMX.

The following example shows how to assign a new throttling policy called “SPIMXPolicy” to user “TCSPIMX”. The new policy explicitly disables throttling mechanisms for EWS connections.

Open the Microsoft Exchange Management Shell and type:

- New-ThrottlingPolicy SPIMXPolicy
- Set-ThrottlingPolicy SPIMXPolicy -RCAMaxConcurrency Unlimited
- Set-Mailbox "TCSPIMX" -ThrottlingPolicy SPIMXPolicy

Note: It can take up to 20 minutes until the new throttling policy is used by all servers.

8.3.2.4 Check MAPI Version Restrictions

Exchange can restrict access for MAPI clients depending on the MAPI version. Make sure that the Exchange server accepts the installed MAPI version (i.e. ExchangeMapiCDO version).

Per server restrictions

The restriction is usually specified per server. Check the following REG_SZ registry value on the Exchange server (if this is a virtual server, check the cluster nodes):

HKLM\System\CurrentControlSet\Services\MSExchangeIS\ParametersSystem\Disable MAPI Clients

If this registry value exists, all MAPI version ranges specified here are disabled.

For example, "-10.6515.6626" disables MAPI clients with versions up to 10.6515.6626 and allows only Outlook XP SP3 or higher. Unfortunately, this value disables ExchangeMapiCDO. Please ask the customer to edit the restriction so that it does not contain MAPI version 6.5.8147.0.

For example, "-5.9.9;7.0.0-10.6515.6626" would be okay. This value disables MAPI versions from 0 to 5.9.9 and MAPI versions from 7.0.0 to 10.6515.6626, and the disabled range does not include ExchangeMapiCDO.

The change will be effective within 15 minutes or to make it effective immediately you can restart the MS Exchange Information Store service.

Per mailbox restrictions

MAPI versions can also be restricted per mailbox. For a newly created mailbox, MAPI access is not restricted. But an administrator might change restrictions later, maybe per batch for all mailboxes in a mailbox store. If TCSPIMX suddenly cannot connect to the server, you may want to check if there is a restriction for the TCSPIMX mailbox: Use the Exchange management shell command Get-CASMailbox and check the property MAPIBlockOutlookVersions. It contains the list of blocked MAPI versions in the same syntax as explained above. The property can be changed via command Set-CASMailbox.

Example: Set-CASMailbox TCSPIMX -MAPIBlockOutlookVersions "5.9.9;7.0.0-10.6515.6626"

8.3.2.5 Handling of Hidden Recipients

Before Exchange 2010, it was possible to use e.g. KCS Voicemail also for Exchange users hidden from the Exchange address list.

Since Exchange 2010, these hidden users are now really hidden. TC/SPI-MX cannot be used anymore for this kind of users.

8.3.2.6 Additional Hints

- ExchangeMapiCDO is not compatible with Microsoft Outlook. Therefore, Microsoft Outlook must not be installed on the system
 - The MAPI component must be installed before installing SPI-MX7.
 - In the case of problems with the MAPI profile, check the registry. Search for "MSMapiApps", usually located at *HKLM\Microsoft\Windows Messaging Subsystem*. Check if the following (reg_sz) key exists and delete it: *HKLM\Microsoft\Windows Messaging Subsystem\MSMapiApps\TCECP.exe*. This key is created e.g. when Outlook was installed before, or if SPI-MX7 setup was started before the MAPI component was installed.
 - When SPI-LN and SPI-MX have to be installed on same PC, then proceed as follows:
 - 1) install SPI-LN
 - 2) install Exchange MAPI
 - 3) install SPI-MX7
- Make sure that TCLINK-MX7 is installed in the environment before starting SPI-MX7 setup.

8.3.3 TTS

It is recommended to install the TTS engine and if necessary the SAPI prior to the KCS setup.

It is possible to install them after the setup but when this is done, the TTSConf utility must be called again after the installation completed.

8.3.4 TC/VoiceLink2

If the VL2 should be used from other mail systems than TCOSS, the MAP file of the used TC/Link must be adapted.

For the TC/LinkSM this adoption would be to copy the lines

```
COMPARESTRING
IEQU
DST.TS_SERVICE
"VOICE_L"

COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_FULLNAME
DST.TS_FREE_ADDR
```

into the entry section

```
"ENTRY ----- two parameters -> service#number -> check language send options"
```

Finally this should look like:

```
ENTRY ----- two parameters -> service#number -> check language send options
*
SET_FREE_ADDRESS
*
SET_FREE_ADDRESS

ISOLATE_LOCALPART
SRC.TS_FREE_ADDR
DST.TS_FREE_ADDR
```

```
REPLACE
DST.TS_FREE_ADDR
REG.TCLSM\Separator
"#"

COPYFMT
DST.TS_FREE_ADDR
"# !"
DST.TS_SERVICE
DST.TS_CORREL_1

SEARCH_ADDRTYPE
DST.TS_SERVICE
DST.TS_CORREL_1
SEP=#

COMPARESTRING
IEQU
DST.TS_SERVICE
"VOICE_L"

COPYAPPEND
DST.TS_FREE_ADDR
SRC.TS_FULLNAME
DST.TS_FREE_ADDR
```

For other TC/Links than TC/LinkSM please consult your technician.

8.4 Setup

Note: The installation and configuration of a VoicexSP scenario is described in [\[10\]](#).

The main entry point to the setup is the dialog to choose which product to install.

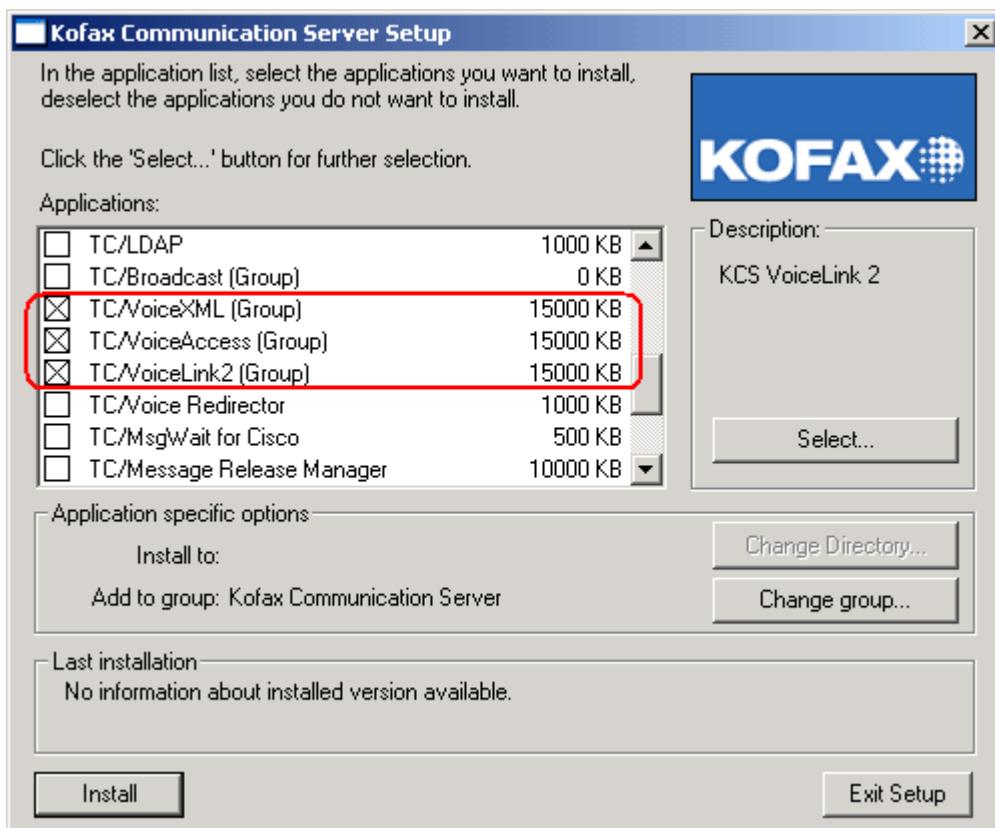


Figure 26 – First Setup Screen

Choose the product to install.

Please verify if all required instances are also checked (Figure 27 – Sample VoiceXSP Instance Selection). To do so, select the product and the press "Select...".

Note: If the setup is actually an update, it can happen that no instance is selected per default. In this case the setup will not install the selected product. So double-check it.

When done, press "*Install*" to start the installation

Because the following installation depends on the chosen products, each product is described separately.

So for the setup of VA go to the subchapter [TC/VoiceAccess](#)

For the setup of VL2 go to the subchapter [TC/VoiceLink2](#)

For the setup of TC/VoiceXML go to the subchapter [TC/VoiceXML](#)

Note: Due to the possible overlapping functionality, some setup screens are only displayed once during setup when installing multiple products.

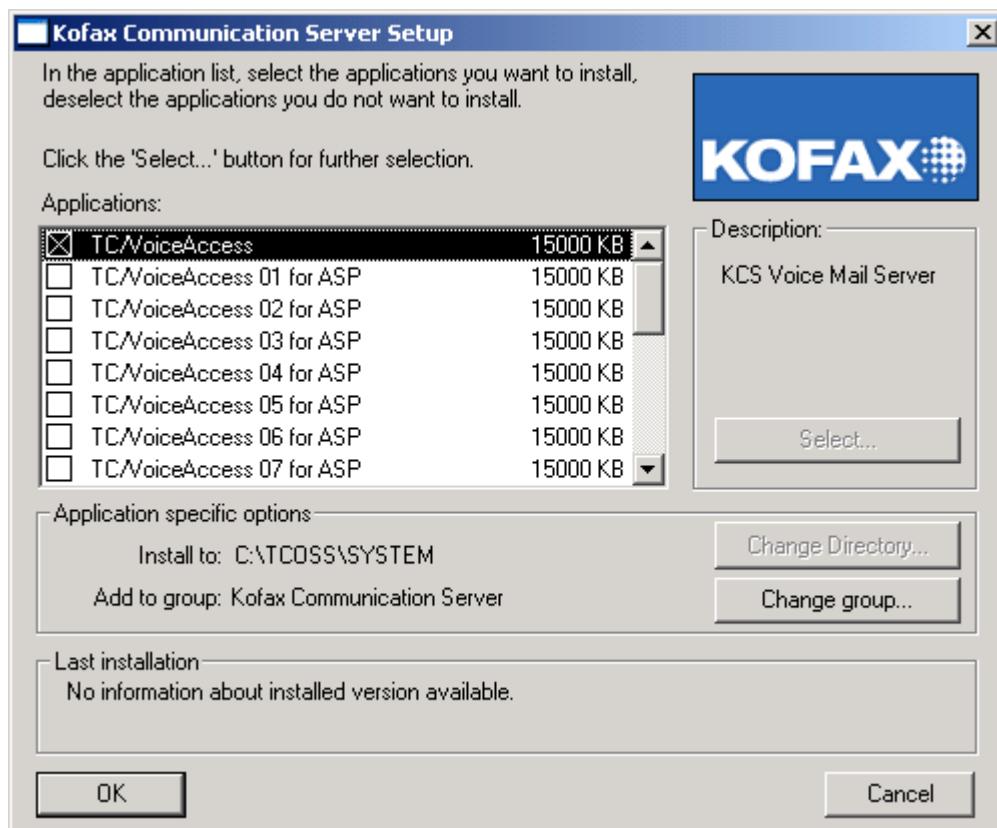


Figure 27 – Sample VoiceXSP Instance Selection

The following installation instructions always refer to one instance. This should not be a problem because all other instances look the same.

Please note that when installing both VoiceAccess and VoiceXML, the following warning might appear:

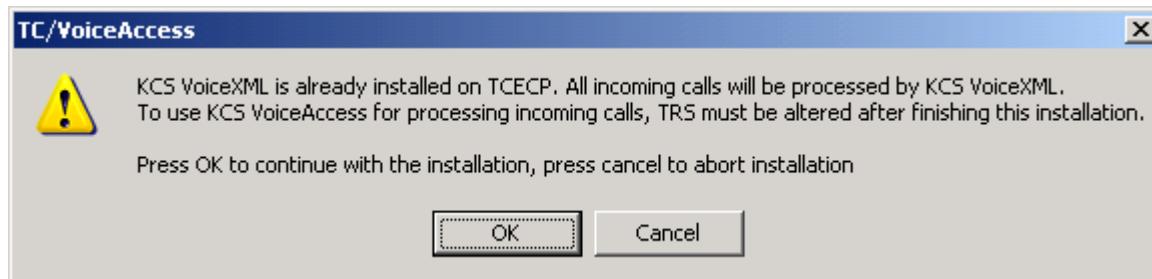


Figure 28 – TC/VoiceXML – TC/VoiceAccess Warning

The reason for this warning is that when TC/VoiceAccess is installed alone, all incoming calls are handled by it. If TC/VoiceXML is installed alone, TC/VoiceXML should handle all incoming calls.

The problem arises if TC/VoiceXML and TC/VoiceAccess are both installed. The setup cannot decide which of the plug-ins should now handle which incoming call. For information about what the TRS is and how to edit it go to the chapter "[TRS – Transparent Routing Script](#)" and "[TCECP - TRS](#)".

8.4.1 TC/VoiceAccess

The first two setup screens for VA represent the basic parameters.

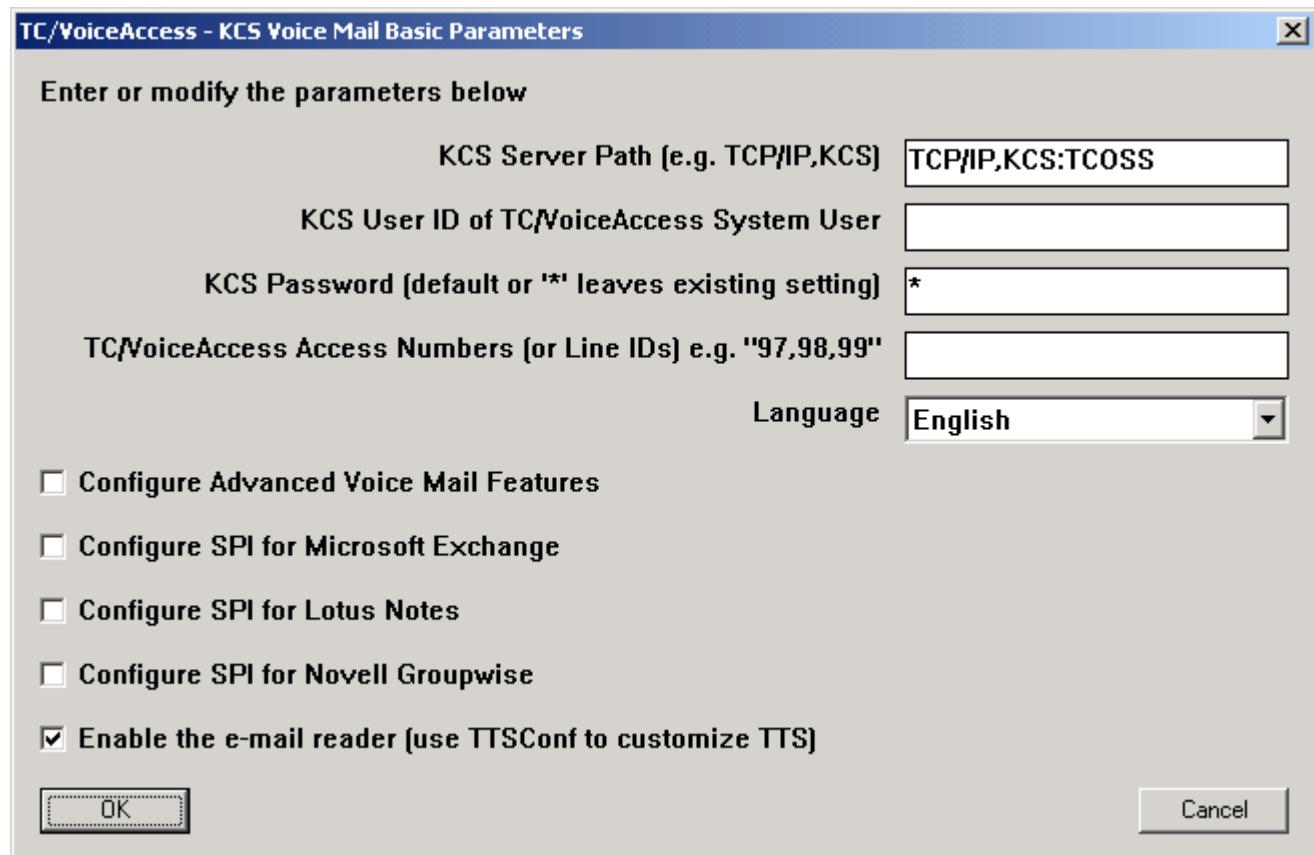


Figure 29 – VoiceAccess Basic Parameters 1

KCS Server Path: Enter the path to the TCOSS server VA should connect to.

KCS User ID of TC/VoiceMail System User: Enter the name of the System User [SysUser]. This user will be used by VA to login to TCOSS.

KCS Password: Enter the password of the SysUser. When updating the system or using the default password, it is not necessary to retype the password. Just leave a “*” in the field and the current password. If currently no password is used, the default password will be used.

TC/VoiceMail Access Numbers: The access number is the number a user dials if he wants to listen to his messages. E.g.: the telephone number of the Voice server is 123 and the access number is 99 then the user has to dial 12399 to hear his messages. For more details see [Different languages for different access numbers](#)

Language: Here a system wide default language can be set. This language will be used, if there is no or a non-existing language is defined by the user or the access number. See [User Depending Languages](#) or [Different languages for different access numbers](#)

Configure Advanced Voice Mail Features: If checked, additional setup screen will be accessible.

Configure SPI for Microsoft Exchange: If checked, additional setup screens regarding the configuration of the SPI will be accessible.

Configure SPI for Lotus Notes: If checked, additional setup screens regarding the configuration of the SPI will be accessible.

Enable the email reader: If enabled, VA will be able to read faxes and emails. After the setup TTSConf must be run. VA can use various TTS engines to read emails and faxes. Per default only the Microsoft TTS engine is installed. For information about TTS see [Supported TTS Engines](#).

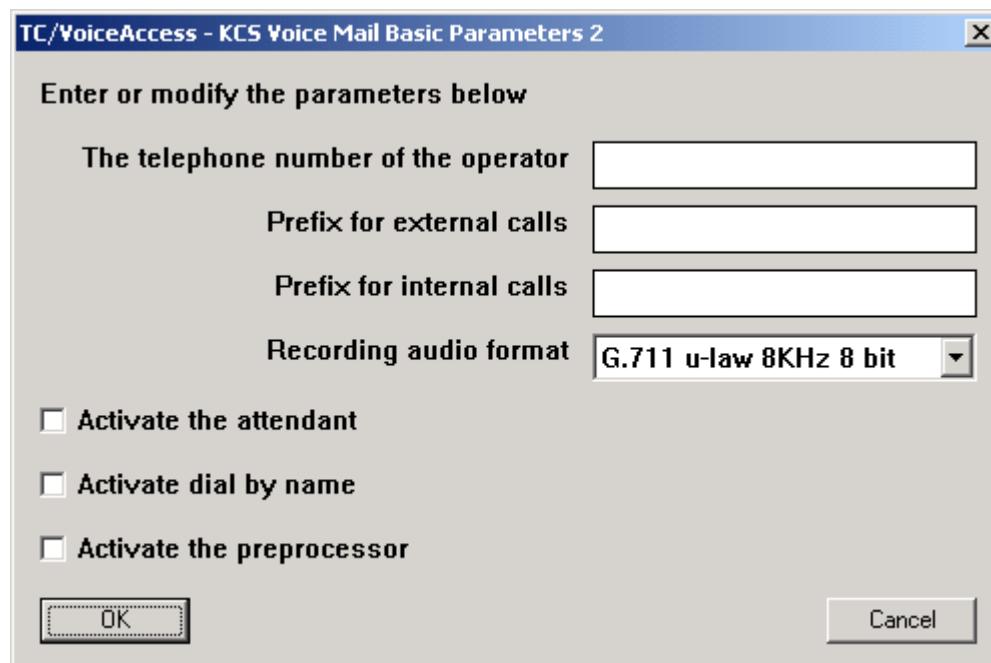


Figure 30 – VoiceAccess Basic Parameters 2

The telephone number of the operator: Enter the telephone number of the default operator which is used when the attendant is activated and no [alternative operator](#) is defined.

Prefix for external calls: Enter the prefix VA needs to get a public line when using [CallSender](#)

Prefix for internal calls: Enter the prefix VA needs to get a line from the telephone system when using [CallSender](#)

Recording audio format: Choose one of the offered recording audio formats. The differences between them are explained in the chapter [Audio formats](#)

Activate the attendant: Enables a more sophisticated menu which allows leaving a message, calling the operator, leaving a fax or using the Dial by name functionality. See [The Attendant](#) for more details.

Activate dial by name: Dial by name (DBN) enables the caller and user to use names of recipients instead of their voice address. See [Dial by Name and Distribution Lists](#) for more details.

Activate the pre-processor: With the pre-processor, VA can resolve Meta-Mail messages and do search and replace in emails.

The next screens will only be shown if the checkbox “activate the pre-processor” is checked.

8.4.1.1 TC/PreProcess

The first TC/PreProcess setup screen contains the settings for *MetaMail*.

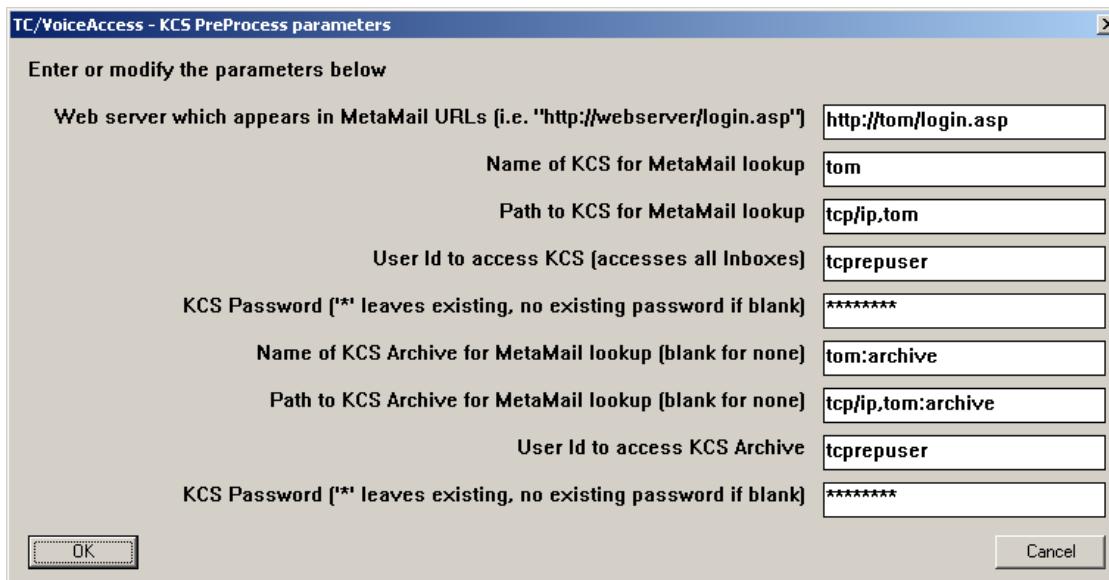


Figure 31 – PreProcessor Parameters 1

Note: The values shown in the screenshot represent a typical example configuration. For a new installation setup suggests no values and all fields are to be left empty.

Web server which appears in MetaMail URLs: This is the host- and domain name of the web server which appears in the *MetaMail URLs*, with preceding “`http://`” and trailing “`/login.asp`”. This string is used to detect *MetaMail URLs*. It is the server where *TC/Web* is running. This field modifies the registry value “`TCPRep\HttpAddress`”. Is must be set equally as the same value on *TC/At2URL* [13] which generates the *MetaMail URLs*.

A server for *MetaMail* lookup should always be entered in the following fields unless the *MetaMail* resolver unit is disabled by setting the registry value “`TCPRep\DoMetaMail`” to 0. Otherwise a delay occurs on start of VA and the *MetaMail* resolver unit will be switched off.

Name of KCS for MetaMail lookup: Name of the Kofax Communication Server where messages specified in *MetaMail URLs* are stored. This is commonly the server where also VA connects to.

Path to KCS for MetaMail lookup: TCTI-path of the Kofax Communication Server where messages specified in *MetaMail URLs* are stored.

User Id to access KCS: A user on the specified Kofax Communication Server with the rights to read all inboxes.

KCS Password: Password of the specified Kofax Communication Server user.

The following settings can be left blank if Kofax Communication Server *Archive* is not used.

Name of KCS Archive for MetaMail lookup: Name of the Kofax Communication Server *Archive* where messages specified in *MetaMail URLs* are stored.

Path to KCS Archive for MetaMail lookup: TCTI-path of the Kofax Communication Server Archive where messages specified in *MetaMail URLs* are stored.

User Id to access KCS Archive: A user on the specified Kofax Communication Server archive with the rights to read the whole archive.

KCS Password: Password of the specified Kofax Communication Server Archive user.

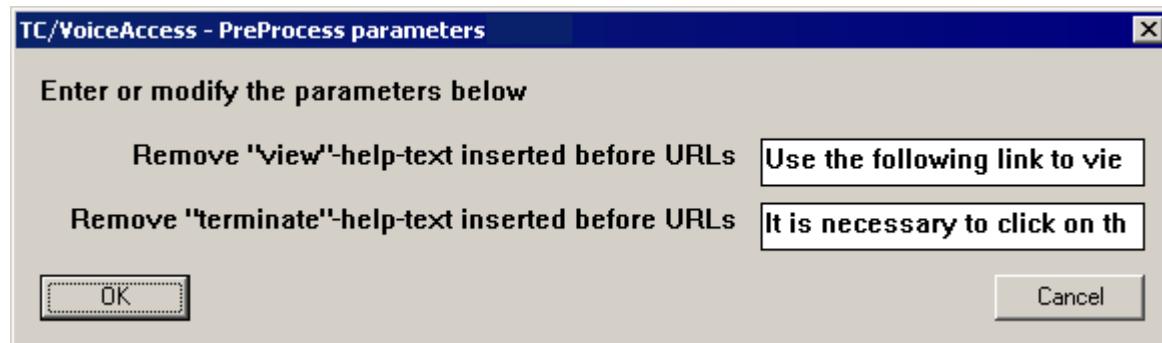


Figure 32 – PreProcessor Parameters 2

The messages which contain the *MetaMail URLs* (*MetaMail messages*) by default contain a help text in the line before the URL. Here is a typical example of such a message.

```
Use the following link to view the attachments of the message:  
http://tom/login.asp?UserId=CHKA&MsgCifId=16512722&MsgId=00FBF6D115566C71&MsgTimeInt  
end=040506:082313
```

When the message is processed by VA it is not desired that this text is read to the user. For this purpose TC/PreProcess has 2 configurable texts which are removed from messages when they appear in a line (without any other text before or after within that line). They must be configured equally as the same values on *TC/At2URL* [13] which inserts the help texts in the messages.

Remove “view”-help-text inserted before URLs: Modifies the registry value “TCPREP\PreURLText”. Setup suggests the default for the same value of *TC/At2URL* [13]; “Use the following link to view the attachments of the message:”.

Remove “terminate”-help-text inserted before URLs: Modifies the registry value “TCPREP\PreURLTextForTermination”. Setup suggests the default for the same value of *TC/At2URL* [13]; “It is necessary to click on the following link to terminate the message:”.

8.4.1.2 TC/VoiceAccess Continued

With the advanced VA setup screens, additional configuration can be done. This configuration is not necessary for a working system.

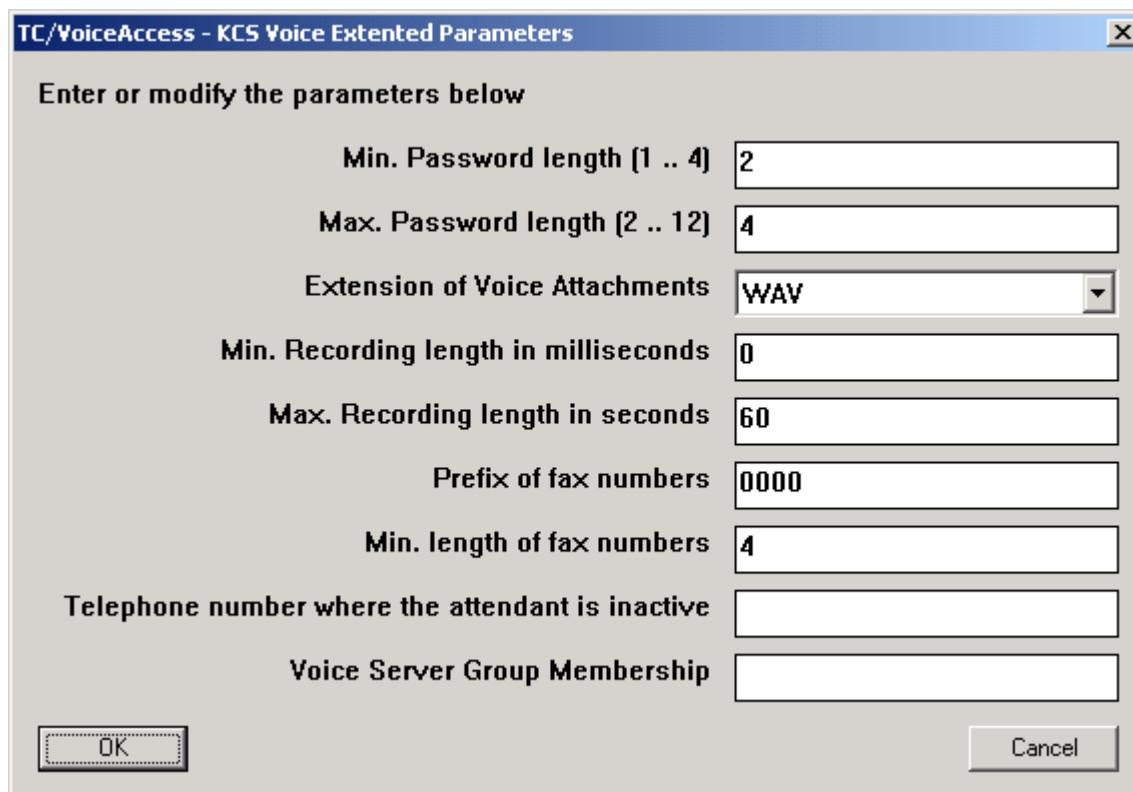


Figure 33 – VoiceAccess Extended Parameters

Min. Password length: Specifies how long at least the password of a mailbox owner has to be.

Max. Password length: Specifies the maximum length of the password of a mailbox owner. See *Password Length*

Extension of Voice Attachment: This option can be “WAV” or “TCS”. The result is that the attachment is named “message.wav” or “message.tcs” within the user’s mailbox. This option has no influence on the recording format. See [Customizable Filenames](#)

Min. Recording length in milliseconds: Enter the minimum length of a voice message. If a caller leaves a messages shorter than this length, it will not be sent. See [Min. Recoding length](#)

Max. Recording length in seconds: Enter the maximum length of a voice message. A caller leaving messages longer than the specified length will be interrupted, and the message will be sent. See [Max. Recoding length](#)

Prefix of fax numbers: If a user composes or forwards a message, he can add this prefix to the recipient’s number. This prefix indicates VA that the entered number may be a fax number. See [Fax and Voice](#)

Min. length of fax numbers: If the user composes or forwards a message and the entered number is longer or equals this value, VA assumes that the number may be a fax number. See [Fax and Voice](#)

Telephone number where the attendant is inactive: Enter the mailbox number where a caller will not hear [the Attendant](#)

Voice server Group Membership: You **must** specify a Group Membership, this is the name of the VSG. (See [Voice Server Groups and Remote Connections](#) for some background information). It is possible to have multiple voice servers in one group so the TC/Player or the TC/VoiceLink2 (if configured) will connect to one of these Voice servers. It is always recommended to connect to a Voice server Group instead of connecting directly to a Voice server, even if there is only one server.

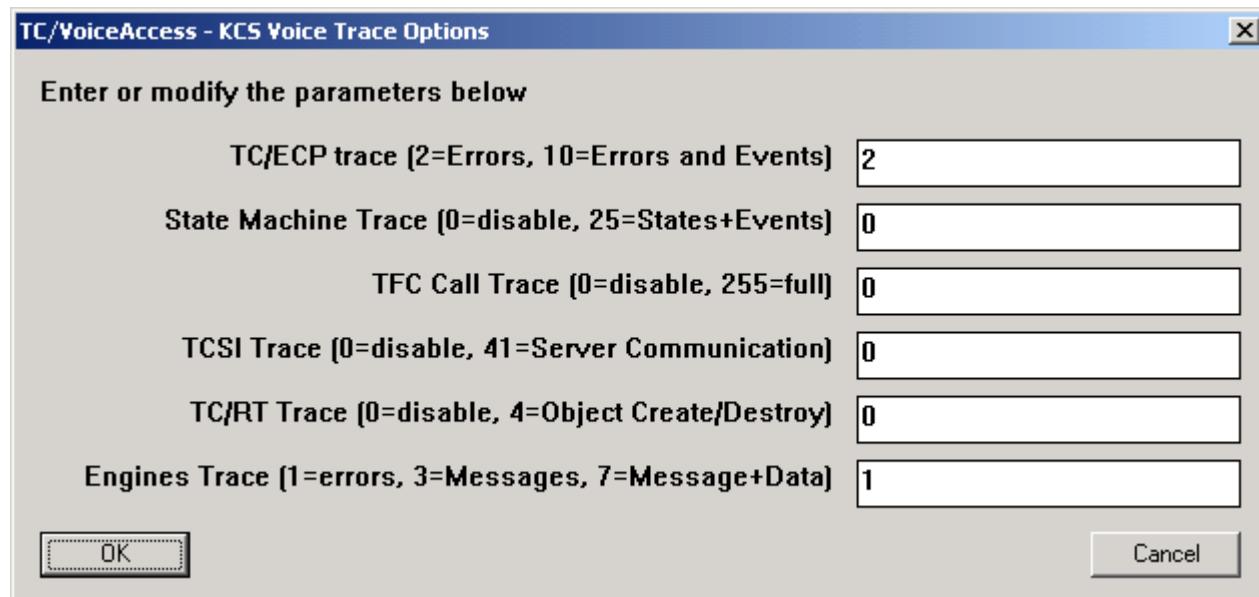


Figure 34 – VoiceAccess Trace Options.

This screen is used to set different trace levels during installation. The values in the fields should only be changed when requested by the support.

The next subchapters will only be used when the corresponding checkboxes have been checked.

- Configure SPI for Microsoft Exchange = TC/SPI-MX
- Configure SPI for Lotus Notes = TC/SPI-LN

8.4.1.3 TC/SPI-MX

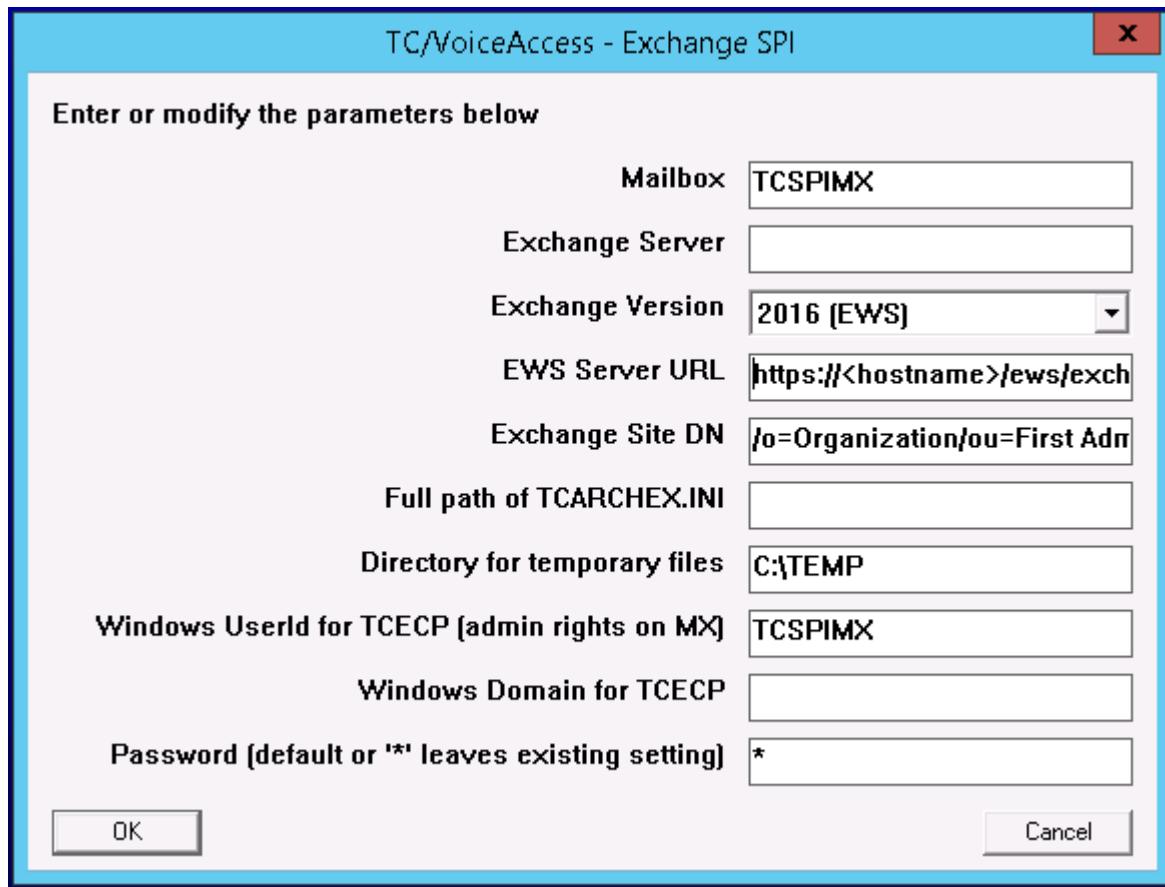


Figure 35 – Exchange SPI 1

TCSPIMX needs a mailbox on an Exchange server and a local MAPI profile to access this mailbox. This mailbox must be created manually, by creating a new Windows user with an Exchange mailbox, or by mailbox-enabling an existing Windows user. TCSPIMX creates a temporary MAPI profile for this mailbox whenever it must log on to Exchange.

Mailbox: Name of the Exchange mailbox used by TCSPIMX. Enter the name of the mentioned Windows user you created before.

Exchange Server: Name of the Exchange server where this mailbox resides. Enter only the server name, without the domain.

Notes:

With Exchange 2010, you have to enter the name of the Client Access Server.

With Exchange 2013, you have to enter the fully qualified internal host name of the Client Access Server.

Exchange Version: Select the desired Exchange Server version and protocol to connect. Following options are available:

- 2007
- 2010
- 2013
- 2013 (EWS)

- 2016 (EWS)

Where, 2007, 2010 and 2013 are based on MAPI. 2013 (EWS) and 2016 (EWS) are based on EWS.

Note: If you are an existing Microsoft Exchange Server 2013 user and need to upgrade to EWS, please reinstall SPIMX and select the “2013 (EWS)” option.

EWS Server URL; In case of EWS protocol, specify the EWS server url.

Where <hostname> is IP Address of Exchange Server.

Exchange Site DN:

This value is composed of the organization name and the name of the administrative group where the server belongs to (e.g., /o=Organization/ou=First Administrative Group).

This value can be determined via the tool TCADUTIL (installed on the link server running dirsnc). Log on to the link server and open a command prompt in folder C:\TCOSS\TCLP. Run tcadutil, specifying the name of the TCECP process user as a parameter. Redirect output to a file.

Example:

TCECP runs under the credentials of user TCSPIMX, who is under the Recipients container.

Run: “tcadutil TCSPIMX >tcspimx.txt”

Open the output file (tcspimx.txt) and locate the line holding the legacyExchangeDN:

legacyExchangeDN,”/o=FSX1/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=TCSPIMX”,true,true,true

The first part of this legacyExchangeDN value (starting with “/o” and ending before “/cn”) is the Exchange Site DN that must be entered in Setup. In this example, the Site DN is:

/o=FSX1/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)

Full path of TCARCHEX.INI: Here you have to enter the full path + filename of a configuration file created by TC/LINK-MX7.

With a default installation, the file name is C:\TCOSS\TCLP\TCARCHEX-TCLINKMX7.INI.

This information is only needed when installing TCSPIMX for the first time.

Directory for temporary files: Path of a directory used for temporary files.

Windows UserID for TCECP: Windows user profile used by TCECP (must have full access to the Exchange mailbox stores). It is highly recommended to specify the same user as specified for “Mailbox”.

Windows Domain for TCECP: domain for this user profile

Password: password for this user profile

Setup now asks whether VoiceMail shall use the standard message classes for Outlook Integration or custom message classes.

A message box gives information about the possible options:

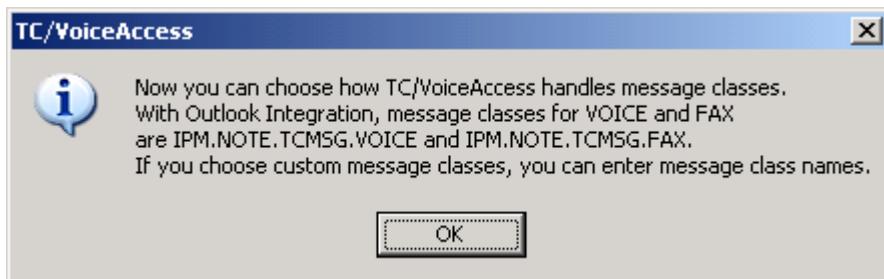


Figure 36 – Exchange SPI Information

Click OK to accept this information. Setup then lets you choose an option

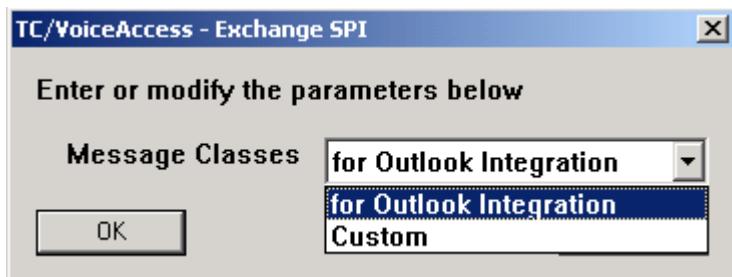


Figure 37 – Exchange SPI Message Classes

If you select "Custom" message classes, you can edit the message class names:

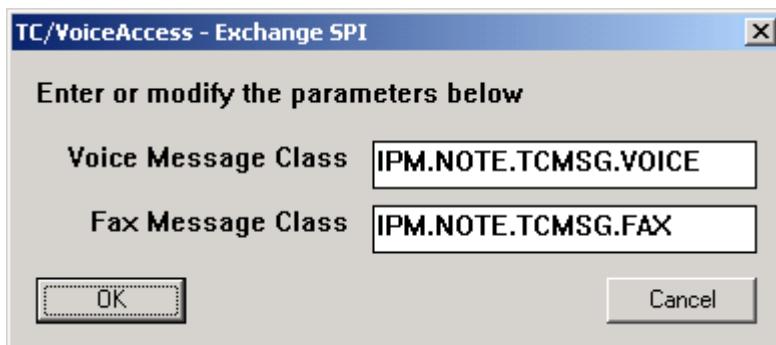


Figure 38 – Exchange SPI Custom Message Classes

Voice Message Class: The name of the message class voice messages have.

Fax Message Class: The class name of fax messages.

8.4.1.4 TC/SPI-LN

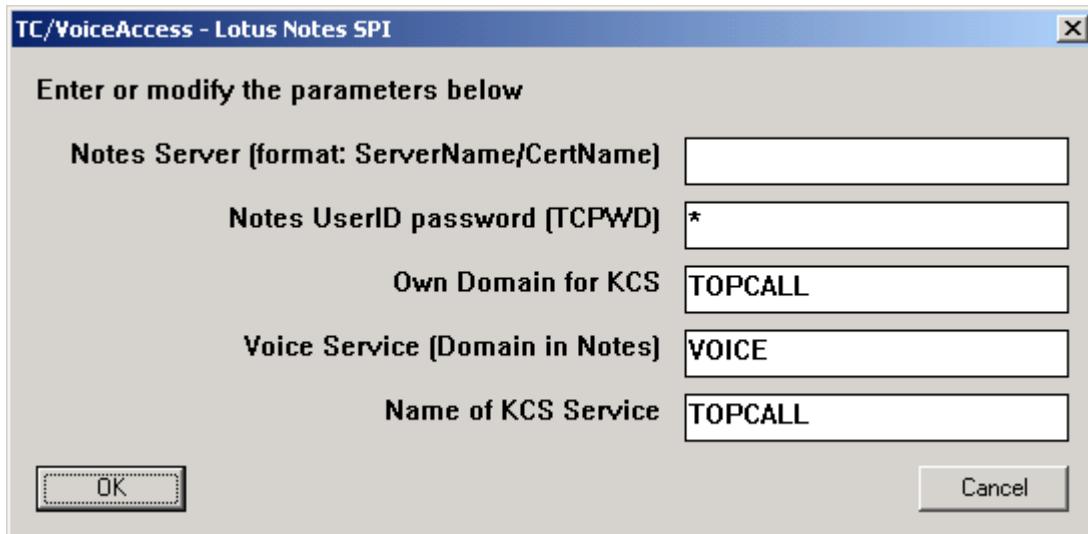


Figure 39 – Lotus Notes SPI Parameters

Notes Server: Default server which is necessary to find the global names & address book.

Notes UserID password (TCPWD): An own module which should enter the password for the client automatically.

Own Domain for KCS: The domain in Notes to send messages.

Voice Service (Domain in Notes): Name of Voice Service

Name of KCS Service: Name of KCS service

NOTE: If this is the first time the TC/SPI-LN is installed on the VS do not forget the [TC/SPI-LN First Time Configuration](#)

8.4.1.5 TC/VoiceAccess Continued

The next two screens can be ignored.

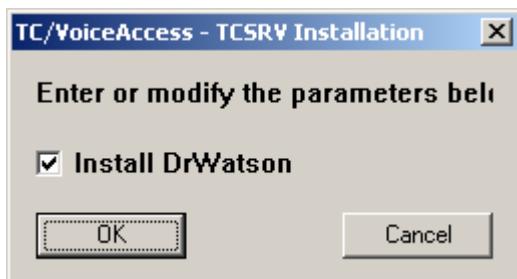


Figure 40 – Install DrWatson for TC/VoiceAccess.

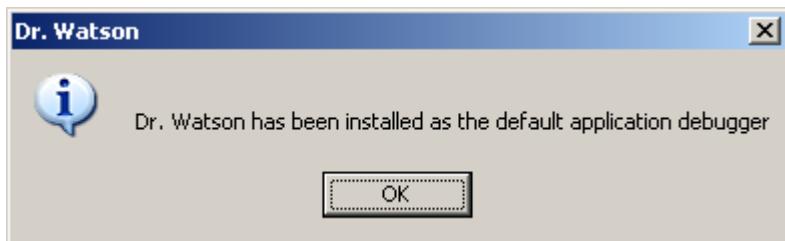


Figure 41 – Successful Installation of DrWatson for TC/VoiceAccess

The next screen is used to setup the VoiceLib.

Warning: If you accept to install the voicelib to its default path (see picture below), the directory “C:\TOPCALL\SHARED\VOICELIB” will be overwritten during setup. All kind of customization will be lost!

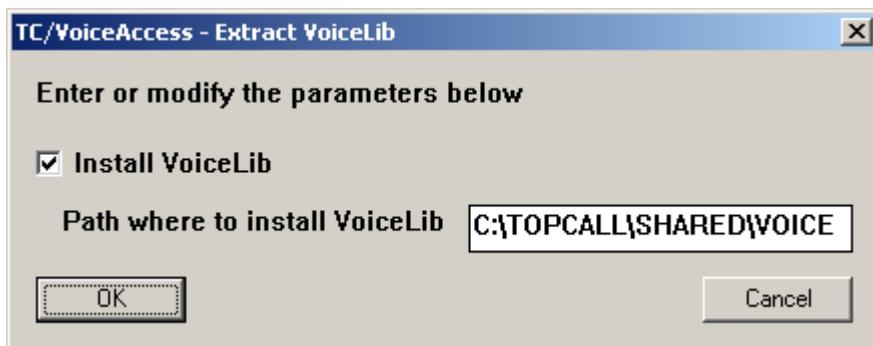


Figure 42 – Install the VoiceLib

Check “*Install VoiceLib*” and click “OK” to copy the VoiceLib to the hard disk.

Now choose the telephone system:

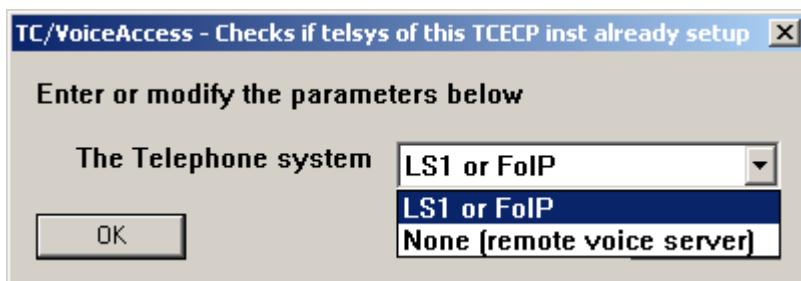


Figure 43 – Choose the Telephone System for TC/VoiceAccess

For more information about integration with line server, see [3]. For more information about FoIPv3, see [23].

At the end you will get asked to run the file “vmaildep.bat” after a restart:



Figure 44 – vmaildep.bat

8.4.2 TC/VoiceLink2

If an update from an existing VL or VL2 should be done, please refer to the chapter [Upgrade Or Downgrade](#).

To install the VL2, select the “TC/VoiceLink2 (Group)” from the main screen. (Figure 45 – VL2 Group Selection)

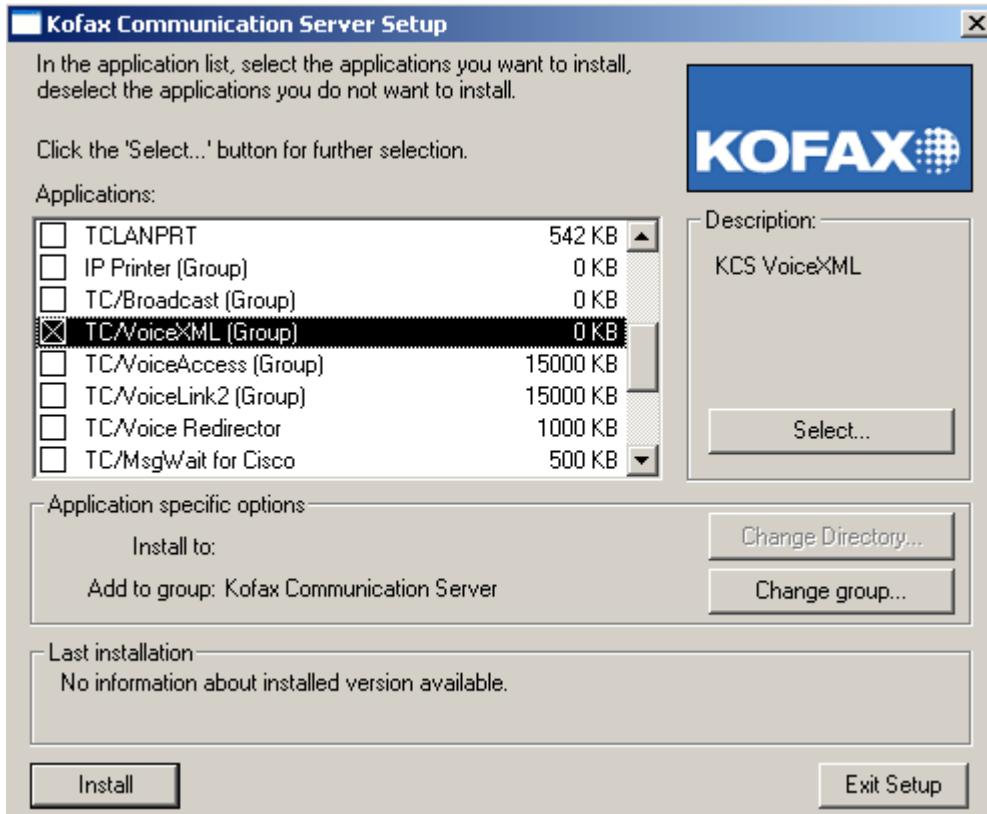


Figure 45 – VL2 Group Selection

To verify if all required instances are selected press “select...” while the “TC/VoiceLink2 (Group)” is highlighted. (Figure 46 – VL2 Instance Selection)

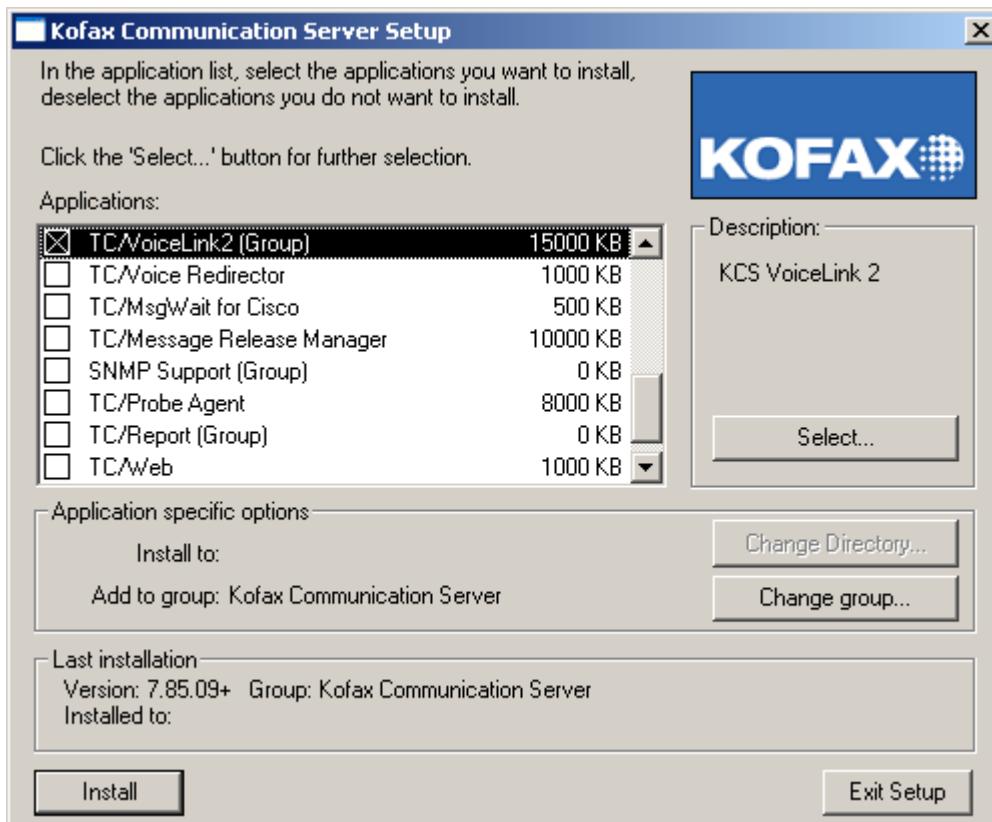


Figure 46 – VL2 Instance Selection

When everything that is required is selected, click on the “*Install*” button.

An upgrade warning may appear (Figure 47 – VL2 Upgrade Warning). This warning should only appear when the VL was already installed.

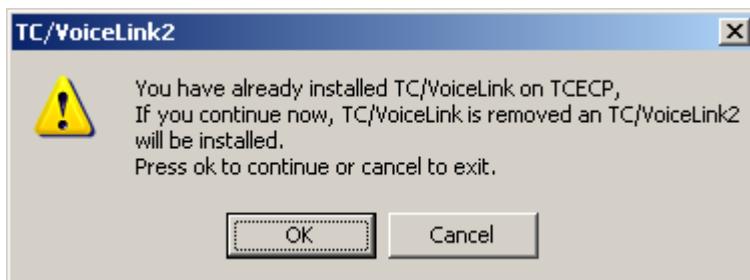


Figure 47 – VL2 Upgrade Warning

The next screen (Figure 48 – VL2 Instances) asks how many concurrent VL2 instances the VS should start.

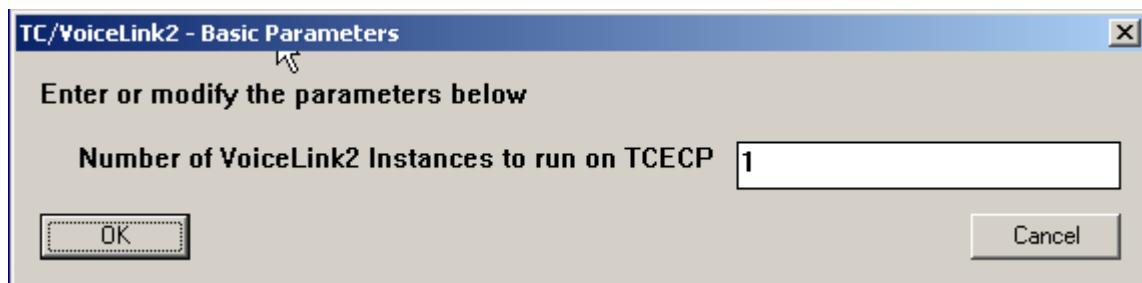


Figure 48 – VL2 Instances

This is the number of concurrent channels the VL2 will use as a maximum. See [Concurrent Voice Channels](#) for information about how to calculate this number.

Enter the name of the licensing server (Figure 49 – VL2 Licensing). The licensing server can be different than the server the VL2 or VA work with (See [TSCI Poller – TCOSS Configuration](#)).

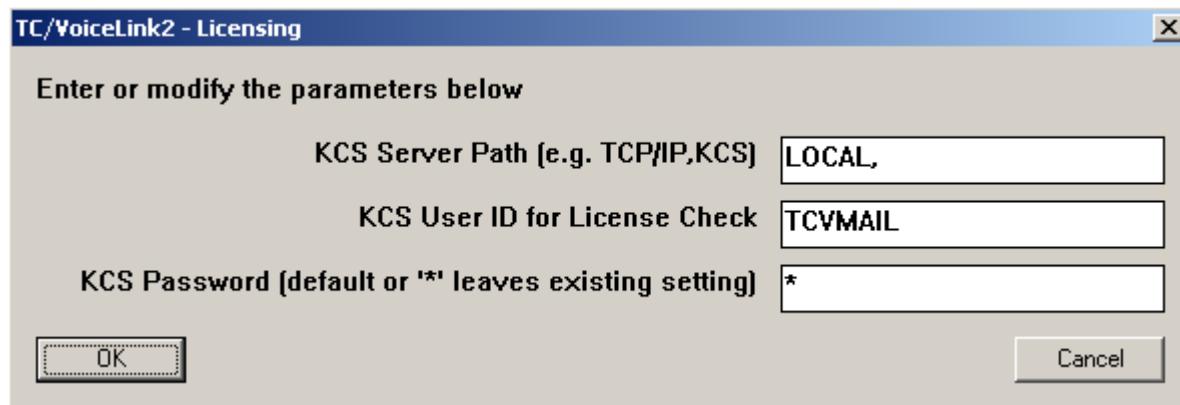


Figure 49 – VL2 Licensing

- KCS Server Path
This is where the licenses for the VL2 are installed.
- Name and Password of VL2 user
The user does not need any rights.

For the next two dialogs, it is save to simply click on “OK”.



Figure 50 – Install DrWatson for VL2

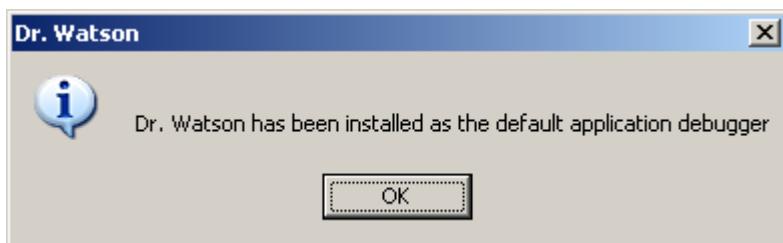


Figure 51 – Successful DrWatson installation for the VL2

A pop-up will ask you to run the file “vlinkdep.bat” after a restart:

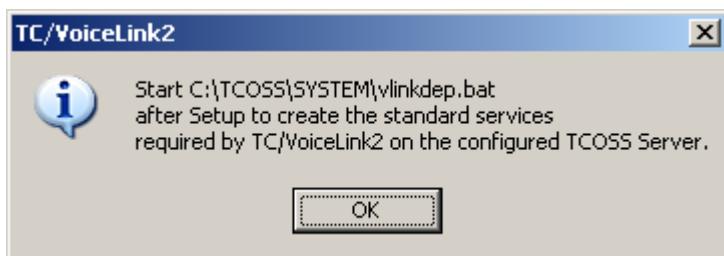


Figure 52 – Vlinkdep.bat

This batch file will create the necessary user and queue “VL2”.

Now choose the telephone system:

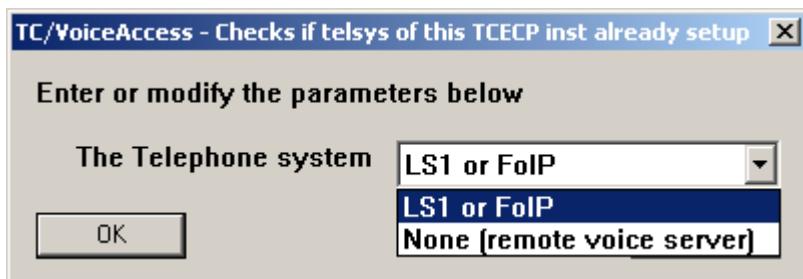


Figure 53 – Choose telephone system for TC/VoiceLink2

- . For more information about integration with line server, see [\[3\]](#). For more information about FoIPv3, see [\[23\]](#).

If this is a new installation, you might be asked to set up TTS for the VL2 (Figure 54 – VL2 configure TTS dialog).



Figure 54 – VL2 configure TTS dialog

After clicking “OK” the setup starts the TTS configuration utility to configure the TTS for the VL2. See [TTS Engine and Speaker Configuration](#) for more information about this.

The tutorial [From VL to VL2](#) does also contain valuable information about the upgrade from the VL to the VL2.

8.4.3 TC/VoiceXML

The TC/VoiceXML installation is rather short and simple.

First, a window asking for the path to the licensing server is displayed.

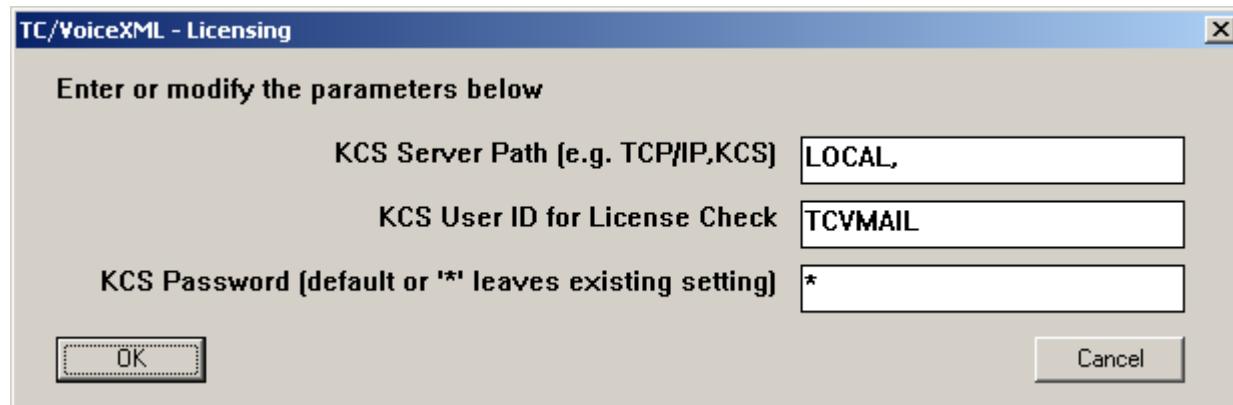


Figure 55 – TC/VoiceXML Licensing

- KCS Server Path
This is where the licenses for the TC/VoiceXML are installed.
- Name and Password of VL2 user
The user does not need any rights.

Then you have to select “Install DrWatson”.

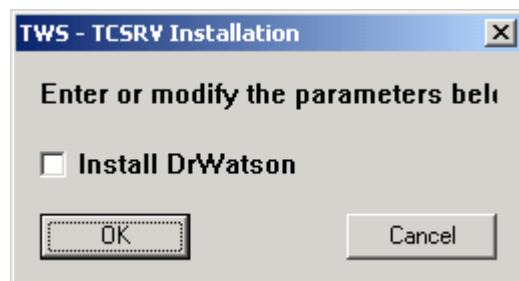


Figure 56 – Install DrWatson for TC/VoiceXML

Afterwards, a confirmation of the successful installation of DrWatson is displayed.

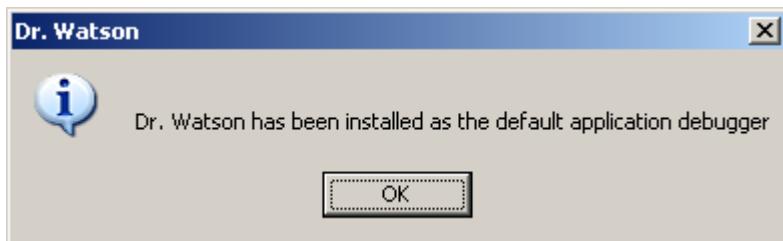


Figure 57 – DrWatson Installed Successfully for TC/VoiceXML

The next step is to select the used telephone system.

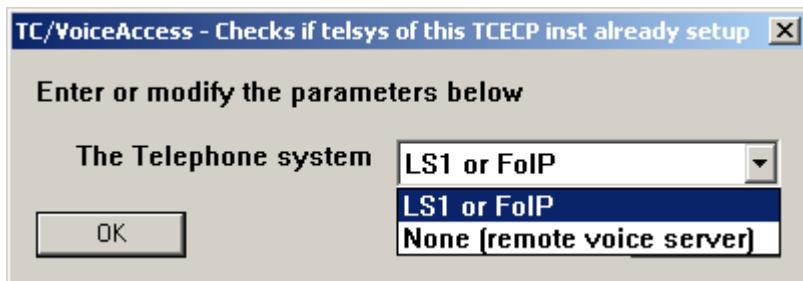


Figure 58 – Choose Telephone System for TC/VoiceXML

- . For more information about integration with line server, see [3]. For more information about FoIPv3, see [23].

Please note that the telephone system setup appears only once per installed TCECP instance. This means that if you install *TC/VoiceAccess 01* and then *TC/VoiceXML 01*, they will both be on TCECP01. Therefore, the telephone system dialogs only appear within the setup of *TC/VoiceAccess 01* but not in the setup of *TC/VoiceXML 01*.

9. Configuration

9.1 General

This chapter provides an overview of the configuration possibilities of a VS. It is split into two main parts. The first part deals with the first time configuration which is absolutely necessary to get a module running. The second part focuses on features and how they are configured. It is arranged in a way that each module describes its features.

- E.g.:
 - A.B.1. TC/VoiceAccess
 - A.B.1.1 A Feature
 - A.B.2. TC/VoiceLink2
 - A.B.2.1. SomeFeature

Note: As mentioned in [How To Read This Document](#) only the relevant parts of a registry key are written.

9.2 First Time Configuration

9.2.1 TC/SPI-LN

The following steps are necessary for the TC/SPI-LN to create a connection between Kofax Communication Server and Lotus Notes.

- After installing KCS, open the Notes client and open the address book. Now you should no longer need to enter a password for the client, so the TCPWD module is installed correctly. If there are problems, make sure the TCPWD.DLL exists in your system and the registry key "NotesAddin\TCPWD" [REG_SZ] is there. Maybe the password is wrong, so re-enter the password in the registry key in plain text, it will be encrypted the next time you open the client (or start voicemail and connect to a Notes user).
- Now make the changes on Kofax Communication Server to access your Notes mailbox with your Kofax Communication Server user:
- Set field "belongs to" to NOTES.
 - 2.) Enter a valid voice address and inactive Notes address.
- Enter a fax password; it is also the voicemail password.
- If the users are created via dirsync, the users should already have an inactive Notes address.

9.2.2 TC/SPI-MX

The following steps are necessary for the TC/SPI-MX to create a connection between Kofax Communication Server and Microsoft Exchange.

- The send as right must be set on the user mailbox.

9.2.3 TCECP

9.2.3.1 Plug-In Configuration

To change the aliases, the number of instances of an active plug-in or which plug-ins are started at all the registry key “General\Plugins” has to be edited.

This key is a “REG_Multi_SZ” where every line is the configuration of a plug-in.

The syntax is:

```
Alias[,Alias[,Alias]]=Instances#[Interface:]Name;Display Name[,Display Name[,Display Name]]
```

The elements of a configuration line are:

Alias: The alias is the name by which TCECP calls the plug-in. It is possible to assign more than one alias to a plug-in. TCECP will remember them and create a new instance of the plug-in as soon as one of this aliases is requested. The TC/Mon32 will display an extra entry for each alias.

Instances: This element is only of interest when configuring an active plug-in. It tells TCECP how many instances should be started during start-up. All those instances will be started with the first alias as name.

Interface: Tells TCECP which interface the plug-in supports. If omitted, it defaults to COM1.

Name: The “true” name of the plug-in. This can be a file name for scripts or a COM-Object name for plug-ins with COM interface. To find the name of a plug-in refer to the [General Module Information](#) chapter.

Display Name: This is the string the monitor will display for this plug-in. More exactly each alias is associated with a display name. If there are more aliases than display names, the last display name will be repeated.

Examples:

```
VoiceMail=TCVoiceMail.VoiceMail/TCVoiceMail.Application;TC/VoiceMail
```

Meaning: VA is a passive plug-in which supports the COM1 interface. It is known to TCECP by the name “VoiceMail” and the TC/Mon32 will display one entry with the name “TC/VoiceMail”.

```
VoiceLink=1#DLL1:c:\tcoss\system\TC\VoiceLink2.dll;TC/VoiceLink2
```

Meaning: The VL2 is an active plug-in with 1 instance. This instance is called “VoiceLink” and gets an entry in TC/Mon32 with the name “TC/VoiceLink2”. The interface type is DLL1.

```
BC-VXML, VOICEXML, ATTENDANT=DLL1:C:\TCOSS\System\TCVXIClient.dll;VoiceBC, VoiceXML
```

Meaning: TC/VoiceXML is a plug-in of type DLL1. It is a passive plug-in. The aliases map to the display names the following way:

“BC-VXML” <-> “VoiceBC”, “VoiceXML” <-> “VoiceXML”, “Attendant” <-> “VoiceXML”

The “Attendant” maps to “VoiceXML” because this is the last available display name.

9.2.4 TCRT

9.2.4.1 Voice Server Groups

When installing a fresh VS into an existing VSG, please enter the name of an existing VS of this VSG into the registry key “RemoteEngine\WellknownServers” [REG_SZ]

9.2.5 TC/VoiceAccess

9.2.5.1 Client Server Channel UC0

VA needs one UC0 Channel for each available telephone channel! Therefore you may need to increase the number of UC0 Channels on the TCOSS server.

E.g. for an ISDN primary rate (E1) line you need 30 UC0 channels. Additional UC0 channels have to be configured for TC/Link and Client applications.

9.2.5.2 Services and Users

During the setup, a popup may appear where you are asked to run the “vmaildep.bat”

This batch file will create the standard services “VOICE”, “VOICE_S” and “VOICE_F”. Additionally the SysUser “TCVMAIL” and the “V” user are created

The SysUser has the default password “tcvmail”.

The SysUser has 2 predefined speech profiles (English and German).

Note: Change the password according to your settings during the setup.

Note: It is not necessary to run the “vmaildep.bat” when the Voice server was only updated.

9.2.5.2.1 Services

VA needs at least 2 services to address and find VA – users. These are the standard services “VOICE” and “TOPCALL”. All the other services are optional. The following table will give an overview and a short explanation of the usage of the different services.

Service Name	Service Type	Prefix	Description
VOICE	Free	V:	The standard Voice service. When a caller tries to leave a message, the recipient must have at least one standard voice address. Voice mails can be left only to users with an inactive VOICE address. Only users with an inactive VOICE address can be addressed when forwarding or composing a message. To change the name of this service, change the value of the registry key: "TCVMAIL\Service [REG_SZ]". For more information about the usage of the service, see VA User settings
VOICE_S	Free	VOICE_S:	Used for the semiautomatic login. To change the name of this service, change the value of the registry key: "TCVMAIL\ServiceSemiAuto [REG_SZ]". For more information about the usage of the service, see VA User settings
VOICE_F	Free	VOICE_F:	Used for the full automatic login. To change the name of this service, change the value of the registry key: "TCVMAIL\ServiceFullAuto [REG_SZ]". For more information about the usage of the service, see VA User settings
TOPCALL	TC		Used to address the recipients of a voice mail. Only users with a TOPCALL service will receive voice mails. To change the name of this service, change the value of the registry key: "TCVMAIL\InternalService [REG_SZ]". For more information about the usage of the service, see VA User settings
VOICE_O	Free	VOICE_O:	This service specifies the telephone number of an Alternative Operator . To change the name of this service, change the value of the registry key: "TCVMAIL\Service_Operator [REG_SZ]". No special configurations are required.

Table 6 – TCOSS Services Used by VA

9.2.5.2.2 SysUser Settings

VA needs one SysUser on TCOSS. In most cases this will be the default SysUser “TCVMAIL” with the password “tcvmail”. The user can be created by executing the batch file “vmaildep.bat”

A SysUser must have at least the following rights:

- Read System Address Book
- Read and Write User Profiles
- List and Open the Inbox of all users
- Enter number directly
- No restriction of services

If you plan to use TTS, the SysUser should have at least one speech profile. For details on how to configure speech profiles, see [\[8\]](#).

Warning: If there is no SysUser or a wrongly configured SysUser, VA will not be able to connect to TCOSS.

9.2.5.2.3 VA User Settings

Most of the possible settings of a user will be described in [8] but there are also some essential settings only the admin can/should do.

Each VA user must have a user profile on TCOSS. Even if the user works with his own mail system (e.g. Lotus Notes) there must be a shadow user profile on TCOSS. Users can either be created with TCfW Communication Server Client or directory synchronization (recommended).

To make a user able to use VA, he must have at least 2 services. "VOICE" and "TOPCALL". (see [Services](#))

The "TOPCALL" service must be active and the address must be the same as the userid. E.g.: The user with the id "SAMY" must have a "TOPCALL,SAMY" address.

The "VOICE" service represents the telephone extension of the user. It will be used as originator of received voice mails (if the originator cannot be identified as TC user). The address of the service has to be a valid telephone number. The service must be inactive. E.g.: A user with the telephone extension 373 must have an inactive address "VOICE,373". This address is used to identify the user if he tries to login to his mailbox (it is his mailbox number), or if somebody wants to leave a message.

E.g.: Somebody is calling into the Voice server 1234-373. Then he hears that he will be connected to the mailbox of the user with the extension 373.

It is possible that more than one user has the same "VOICE" address. In this case, the first one will receive the message. The first one is defined as the user with the lowest case insensitive user id (short name).

E.g.: If the user "Hu" und "Nu" have the same "VOICE" address, "Hu" will be used first.

It is also possible that one user has more than one "VOICE" address. This enables him to receive voice mails from different telephone extensions into one mailbox.

Additionally the following properties of the user profile are used:

- The System Address Book entry is used as originator of received voice messages.
- From the "Manual Fax" panel within the TCfW user profile, the "Access password" is used as VA password
- The field "User Belongs to" indicates the location of the mailbox (Lotus Notes, MS Exchange, or TOPCALL).

Optionally each user can have additional services.

The service "VOICE_S" is used for semiautomatic login. The semiautomatic login will be done when the "VOICE_S" address matches the caller's telephone number or extension.

The difference between standard or semiautomatic login is that VA prompts only for a mailbox password instead of additionally asking for the mailbox number.

E.g.: A caller with the extension 192 calls the access number of the Voice server. The Voice server finds the match "VOICE_S,192" and only prompts the caller to enter his password.

The service "VOICE_F" works the same way like "VOICE_S" only that instead of a semiautomatic login a full automatic login is done.

The difference between standard or full automatic login is that VA will not prompt the mailbox owner for his mailbox number or password.

E.g.: A caller with the extension 192 calls the access number of the Voice server. The Voice server finds the match “VOICE_F,192” and allows the caller to enter the mailbox without prompting for mailbox number or password.

With the service “VOICE_O” the administrator can specify one [Alternative operator](#).

9.2.5.2.4 Shadow Users

A shadow user has the same settings like a standard VA user. The 2 differences are the “User belongs to” field and one additional inactive address.

The “User belongs to” field shows a list of mail systems (Notes, Exchange) which can be used for TC/SPI.

According to the “belongs to” field the user needs one inactive address, the proxy address. The address consists of the SPI’s service e.g.: “NOTES” and the address of the user in the foreign mail system. E.g.: CN=Lionel North/O=VOICE.

If the user belongs to Lotus Notes, a proxy address with service NOTES and Notes user Id must be defined (e.g. NOTES,CN=Lionel North/O=VOICE).

If the user belongs to Microsoft Exchange, a proxy address with service MX7 and the first email address of the user must be defined (e.g. MX7,user97@devtest.com).

Note: For the last two points this is the same as it is used today for advanced originator mapping

9.2.5.3 Kofax Communication Server

VA uses the following settings to connect to a Kofax Communication Server (All registry keys are of type [REG_SZ]):

“TOPCALL\Path” [REG_SZ] is the path to the Kofax Communication Server (e.g.: “TCP/IP,MYServer”)

“TOPCALL\Server” [REG_SZ] is the name of the Kofax Communication Server (e.g.: “MyServer”)

“TOPCALL\User” [REG_SZ] is the name of the VA SysUser. (per default “tcvmail”)

“TOPCALL\Internal” [REG_SZ] the encrypted password of the SysUser.

9.2.6 TC/PreProcess

9.2.6.1 TCOSS Server and User

If *MetaMail* support is enabled a TCOSS server and user with the right to read all users inboxes must be configured on the TCOSS server. If an archive server is also used, another user must be available on the archive server.

	Group members			All users		
	List	Correct	Open	List	Correct	Open
Inbox	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Outbox	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Message Folder	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

The servers and users must be entered in the [first TC/PreProcess setup screen](#).

9.2.6.2 TFC License on TCOSS Server

For access to the TCOSS and archive servers for *MetaMail*, a TFC license must be installed on these servers.

9.2.7 TC/VoiceLink2

9.2.7.1 Setup, Users and Queues

After the VL2 installation, it is recommended to execute "C:\TCOSS\SYSTEM\VLINKDEP.BAT". This batch file creates the queue (the user) "VL" on the TCOSS server.

The user "VL" is also the queue which is polled by the VL2. He does not need any specific rights and no special addressing. Actually the user "VL" does not need any rights as this does also work:

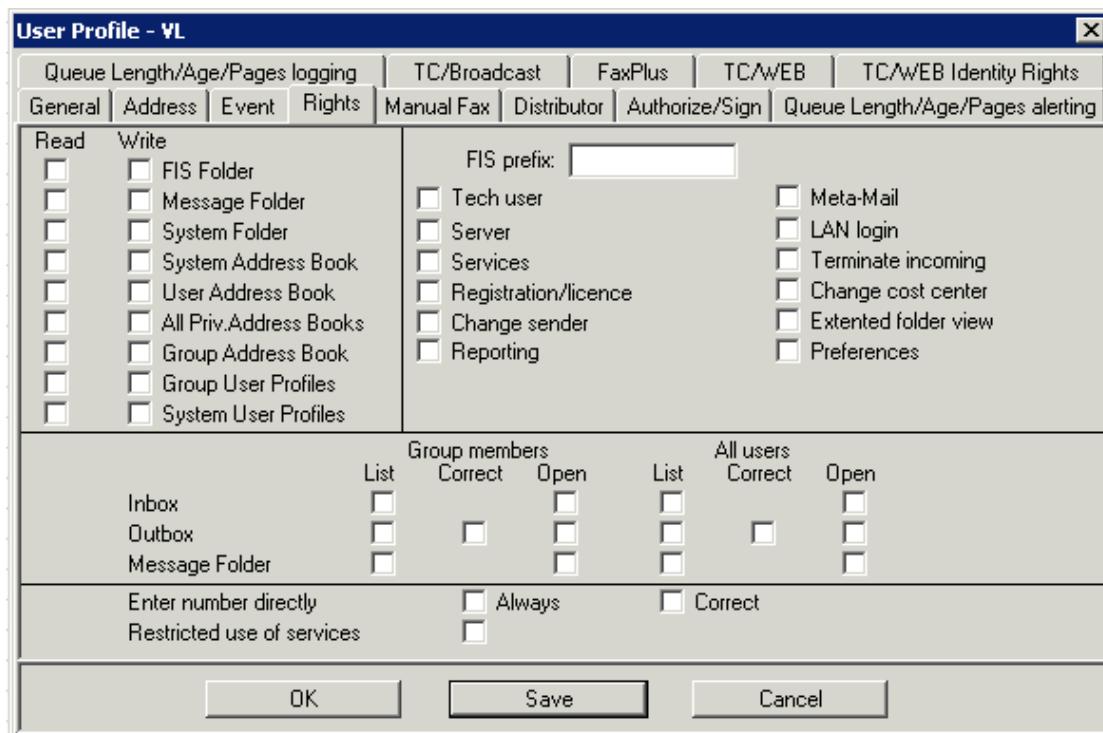


Figure 59 – User VL Rights

It is recommended to monitor the inbox of the user "VL" to terminate open messages manually.

The default password of the user “VL” is “password”.

9.2.7.2 The Kofax Communication Server

It is only necessary to configure a Kofax Communication Server if the TSCI poller or TSCI logger is used. In both cases these modules will automatically use the Kofax Communication Server which is configured below the “TOPCALL” sub key. If a different Kofax Communication Server is required please see chapter [TSCI Poller – TCOSS configuration](#) and [Logging & Billing](#).

The configuration of the TCSI codepages is described in [TCSI Codepage Conversion](#) for additional information.

9.2.7.3 The Licenses

Check if there is at least one VoiceXML and VL2 license left to be used.

9.2.8 VoiceXSP

Please refer to [\[10\]](#)

9.3 Features

9.3.1 TCECP

9.3.1.1 TC/Mon32 Output

The output TCEP produces on the TC/Mon cannot be configured.

Basically there are two different plug-in types and therefore two different ways the status is displayed.

A typical status display for a passive plug-in (e.g.: VA) would look like this:

```
... VoiceAccess: Connections active(redirected): 0(0)  
... Errors: 0 Warnings: 0
```

Figure 60 – TC/Mon32 Output for VA

The first line shows the current activity of the plug-in.

- The first digit is the count of active calls the plug-in is serving.
- The second digit is the count of session which are redirected to the plug-in. This count is not included into the count of active calls.

The second line shows the current count of errors and warnings the plug-in encountered. This line does only show when an error or warning occurred.

The status display for an active plug-in (e.g.: "TC/VoiceLink2") would look like this:

```
... VoiceLink2: available/active(redirected): 1/0(0)  
... Errors: 1 Warnings: 1
```

Figure 61 – TC/Mon32 Output for VL2

Again, the first line shows the current activity of the plug-in.

- The first digit is the count of started and currently available instances of the plug-in.
- The second digit is the count of instances that are currently processing a call. If this digit is zero all instances are ready. The count of active instances may not exceed the count of available instances.
- The third digit is the count of sessions which are redirected to the plug-in. This count is not included in the count of active instances but depending on the plug-in this may vary.

The second line shows the current count of errors and warnings the plug-in encountered.

9.3.1.2 TRS

The TRS has already been discussed in [“TRS – Transparent Routing Script”](#).

For more information about how a TRS is written, check [\[1\]](#) or read the original “trs.vbs” supplied with the VS under “C:\TCOSS\System”

9.3.1.2.1 Configuration

The file name of the TRS can be configured with the registry key “General\TRS” which is of type REG_SZ.
The default is “C:\TCOSS\System\trs.vbs”

9.3.1.3 Delayed Active Plug-In Start

Because of active plug-ins may load longer and more if a new instance is created, TCECP offers the possibility to add a delay between each instance creation.

This avoids performance bottle necks and minimizes the risk to interfere with already running VoiceXSP instances.

9.3.1.3.1 Related Topics

[Active Vs Passive Plug-in](#)

9.3.1.3.2 Configuration

The registry key “General\ActiveCallsDelayStart” [REG_DWORD] controls the delay between the creations of the instances in msec.

9.3.1.4 Maximum Memory Usage

It may happen that the memory consumption of the VS increases over time. this feature was implemented to prevent a standstill because of missing memory. If activated, the VS checks periodically how much memory is used and does a restart if necessary.

It is noteworthy that the VS checks the global memory situation. If a second program consumes all available memory on the server, TCECP will keep rebooting.

The current memory usage in percent is calculated with the following formula:

$$\frac{(\text{Total phys Memory} + \text{Total PageFile} - \text{Available phys. Memory} - \text{Available PageFile})}{(\text{Total phys Memory} + \text{Total PageFile})} * 100$$

Note: The shutdown TCECP will perform if the memory limit is reached is no [graceful shutdown](#).

9.3.1.4.1 Configuration

The maximum used memory can be configured with the registry key “General\MemoryUsageMax” [REG_DWORD]. The value is a percentage after which the VS restarts. When set to 0, this feature is disabled.

9.3.1.5 Performance Counters

The VS does not only provide statistical information through the TC/Mon32 but is also capable of doing this via performance counters. This performance counters can then be viewed via the performance monitor ("perfmon.exe").

It may be interesting to have a look at the performance counters because they provide additional information to the already available in TC/Mon32. One point is that plug-ins can create their own performance counters which are not displayed in TC/Mon32.

Per default available performance counters are:

- The number of errors occurred per alias.
- The number of warnings occurred per alias.
- The average invocation duration in (msec) of an alias.
- The number of active instances of an alias.
- The number of redirected calls to an alias.

9.3.1.5.1 Related Topics

[TC/Mon32 Output](#)

9.3.1.5.2 Configuration

The registry key "EnablePerformanceCounters" [REG_DWORD] has to exist and must be set to "1".

When calculating average values, TCECP does not calculate them over all previous values but only over a certain amount. This amount can be configured with the key "General\AveragingBacklog" [REG_DWORD] which is per default 10.

9.3.1.6 Timeouts

TCECP does not only create instances it also monitors them. There are several different timeout values which apply to different aspects during the lifetime of an instance. But there are also timeouts for the shutdown process.

9.3.1.6.1 Prerequisites and Additional Installation

Tcvsvr.exe version 7.13.02 or higher (part of KCS 7.65.28 or higher)

9.3.1.6.2 Related Topics

[Graceful Shutdown](#)

9.3.1.6.3 Configuration

Following timeouts can be configured:

- Worker: It occurs when a plug-in does not respond to an event (E.g.: the user pressed a key after the given time. The registry key is "General\WorkerTimeout" [REG_DWORD] in msec.
- Session: The session timeout happens when TCECP is waiting for TCRT activity. Such an activity could be a DTMF or the indication that the user disconnected. The registry key is "General\SessionTimeout" [REG_DWORD] in msec.

- Console: Is not a true timeout but controls the interval in which TCECP checks for the [Maximum Memory Usage](#). Additionally, it signals that the main thread is still alive. The registry key is "General\ConsoleTimeout" [REG_DWORD] in msec.
- Supervisor: Occurs if the main thread of TCECP hangs. The registry key is "General\SupervisorTimeout" [REG_DWORD] in msec.
- Graceful shutdown: Is the amount of time TCECP will wait for a graceful shutdown before it proceeds with a normal shut down. The registry key is "General\ GracefulShutdownTimeout" [REG_DWORD] in sec. When increasing this timeout, check if it still fits into the stop process timeout of the TCSRV.
- Shutdown: Is the amount of time TCECP will wait for any plug-in to close before forcefully shutting down. The registry key is "General\ ShutdownTimeout" [REG_DWORD] in sec. When increasing this timeout, check if it still fits into the stop process timeout of the TCSRV.
- TCSRV – stop process timeout: This timeout is active when a "graceful shutdown" or a "kill process" is issued via TC/Mon32. After the specified time elapsed, the TCSRV will terminate the process. **Always check that this timeout is bigger than the timeout for the graceful shutdown plus the normal shutdown timeout.** The registry key is "ProcessStopTimeout" [DWORD] in seconds where the default is 60.

Attention: The supervisor timeout must not be smaller than the console timeout!

9.3.1.7 Graceful Shutdown

In some cases the VS has to be restarted during production hours where calls are processed. In such a case it is not feasible to shut down the server and terminate the calls.

The solution is the "graceful shutdown" mechanism. When a graceful shutdown is issued, the VS tells this every plug-in and rejects incoming calls. A plug-in that receives a graceful shutdown request can decide if it simply proceeds or it can take measures to close the call as soon as possible.

TCECP does not wait until all calls are finished but after a timeout (see [Timeouts](#)), it will proceed with the normal shutdown. Again the plug-ins are informed about the shutdown and again, a timeout takes care that it does not take too long. If, after the last timeout, a plug-in is still running it gets terminated.

The graceful shutdown can be issued either by pressing "s" in the console window or by choosing "Stop Process" in TC/Mon32 (see Figure 62).



Figure 62 – TC/Mon32 Graceful Shutdown.

9.3.1.7.1 Prerequisites and Additional Installation

The TC/Mon32 version must be at least 7.13.00

The TC/SRV version must be at least 7.13.00

9.3.1.7.2 Configuration

See [Timeouts](#) for details.

9.3.1.8 Dynamic Reconfiguration

The dynamic reconfiguration allows the administrator to control when TCECP and its plug-ins will reload their configuration.

A reconfiguration can be issued either by pressing "r" in the console window or by choosing "Reload configuration" in TC/Mon32 (see Figure 63)



Figure 63 – TC/Mon32 Dynamic Reconfiguration

TCECP itself will reload:

- The trace level
- The timeouts
- The maximum memory usage
- The performance counter settings

It will **not** reload:

- The plug-in configuration (alias, active instances...)
- Anything related with TCRT. TCRT does reload certain parts on its own. TCECP does not have any influence on that.

After TCECP reloaded its settings, it forwards the request to every plug-in that supports the DLL1 interface. Then the plug-in to reloads its own settings.

Each plug-in should provide information about what is reloaded in the [General Module Information](#) chapter.

9.3.1.8.1 Prerequisites and Additional Installation

The TC/Mon32 version must be at least 7.13.00

The TC/SRV version must be at least 7.13.00

9.3.1.8.2 Related Topics

[Maximum Memory Usage](#)

[Performance Counters](#)

[Timeouts](#)

[General Module Information](#)

9.3.1.8.3 Configuration

No configuration necessary.

9.3.1.9 Tracing

The trace level can be changed with the registry key "General\Tracelevel" [REG_DWORD].

It is a bit field with following supported values.

- 0x02 = Error trace which writes also event logs
- 0x04 = Warning trace
- 0x08 = Operational trace
- 0x10 = Function call trace
- 0x20 = Object trace.
- 0x40 = Reference counters. Construction / Deconstruction

For debugging reasons it is recommended to set the trace level to “0xff”.

9.3.2 TCRT

9.3.2.1 Global TTS (De)Activation

In certain situations, the usage of TTS is not necessary. If the TTS is deactivated every attempt to use this service will lead to silence on the phone.

9.3.2.1.1 Related Topics

[Disable TTS \(TC/VoiceLink2\)](#)

[Supported TTS Engines](#)

9.3.2.1.2 Configuration

Set the registry key “TCRT\ResEnableTTS” [DWORD] to “0” to disable TTS support. Set the key to “1” to enable TTS support.

9.3.2.2 TTS Engine and Speaker Configuration

The voice server supports the Microsoft Text-to-Speech API Version 5 (Microsoft SAPI 5 Interface) beside the SAPI 4 Interface. It is possible to use Text-to-Speech engines that support either the Microsoft SAPI 4 or SAPI 5 Interface. SAPI 4 and SAPI 5 speakers must not be mixed.

TTS engine vendors often support both, the Microsoft SAPI 4 and the SAPI 5 Interfaces. However, the support for SAPI 4 is ceasing; especially newer versions of TTS engines are released first with SAPI 5 support. Additionally SAPI 5 has a better performance, the internal handling of the configuration and the interface is easier, and the overall stability is better than with SAPI 4.

9.3.2.2.1 Prerequisites and Additional Installation

SAPI 4 and/or SAPI 5 installed.

At least, one TTS Engine installed. (The MS-TTS is installed when installing the SAPI)

The TTS support must be enabled.

9.3.2.2.2 Related Topics

[Global TTS \(De\)Activation](#)

SSML Support

9.3.2.2.3 Configuration

Start the TTSConf tool. It is normally located under “Programs -> Kofax Communication Server -> TTS Configuration Tool”.

Note: The VS must not run while configuring the TTS Engines and Speakers.

After startup the following screen (Figure 64 – TTSConf Startup Screen) should appear.

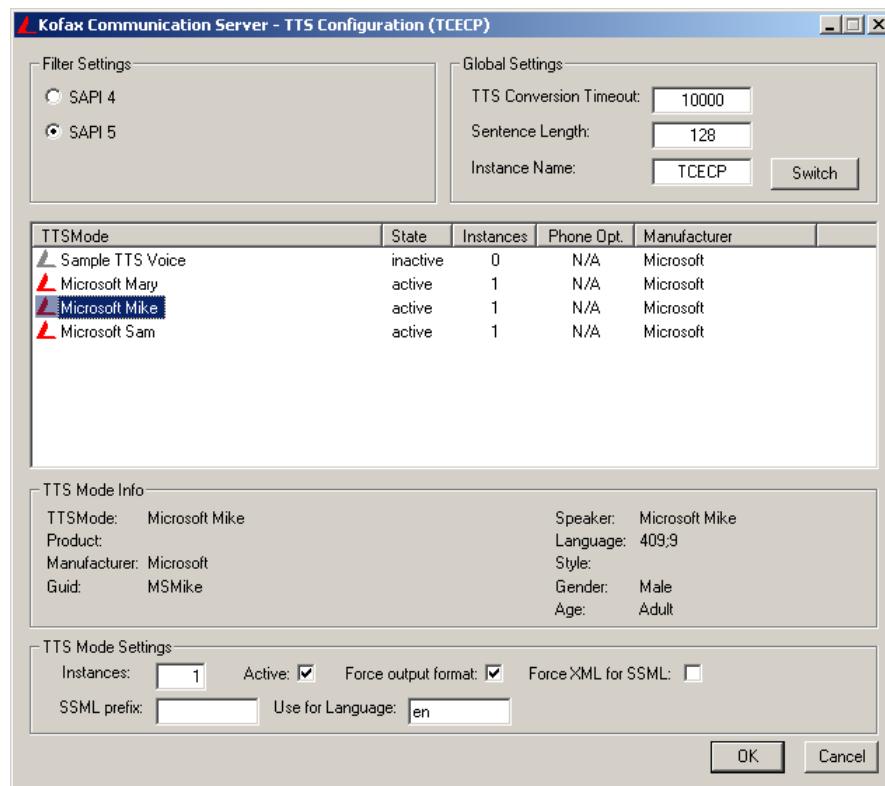


Figure 64 – TTSConf Startup Screen

In the top-left “Filter Settings”, you can choose the version of the Microsoft SAPI Interface. This is either SAPI 4 or SAPI 5. Depending on the chosen interface, you will see a list of either the SAPI 4 or the SAPI 5 TTS speakers installed on the system. **You can only use the speakers of one interface!** However, typically a TTS engine now supports either SAPI 5 or both SAPI versions.

The rest of the TTS configuration tool works as the previous version. Choose the speaker, configure the number of instances you want to use (this has to be according to the TTS channel licenses you purchased), and set the speaker to active. The icon changes from grey to red. By clicking the “OK” button the configuration is saved. That means the active speakers of the chosen interface (either SAPI4 or SAPI5) and their configuration are written to the registry. It is not necessary to deactivate the speakers of the unused interface.

When starting the VS the configured number of instances of the chosen TTS speakers (of the configured interface) will be started and initialized.

If one of the speakers cannot be started or initialized, the VS does not start up and an entry will be written to the Windows Event-Log (“*Error in plugin TCE_RES (Code >Init TTS-Engines Failed<) Check plug-in documentations for detailed error reason.*”). If TTS is configured, but none of the speakers are active, the VS will not start but write a Windows Event-Log message (“*Error in plugin TCE_RES (Code >No TTS-Engines Loaded<) Check plugin documentations for detailed error reason.*”).

9.3.2.3 Languages

The VS can handle multiple languages at the same time. It is even possible for the VS to switch languages during a call.

At the moment the VS knows two ways to handle languages.

- VS uses speech profiles and TC language ID's.
- The XML way: The VS uses the “xml:lang” attribute. E.g.: “en-US” for US English.

9.3.2.3.1 TC Languages

TC Languages are defined by a two-digit-number. E.g.: “01” would be English. “02” would be German. See Table 7 – TC Languages for a complete list.

ID	Language
01	English
02	German
03	French
04	Custom Language 1
05	Custom Language 2
06	Custom Language 3
07	Japanese

Table 7 – TC Languages

This list is not VS specific but used by all KCS products, e.g.: When changing the language of a Kofax Communication Server user with TCfW.

If the desired language is not listed, it is highly recommended to use one of the “Custom Languages”. This is because otherwise other TC products may have trouble. E.g.: It is not possible to change the language with TCfW Communication Server Client to something bigger than “07”.

The VS uses the language ID mainly in conjunction with the “Language ReSource” [LRS] Files. These files allow a simple internationalization within a voice application. The application only has to specify what to say and the VS works out how to say it. This concept is not limited to pre-recorded prompts but can also be used for TTS output. For more details about using and programming LRS files, please refer to [\[1\]](#)

Because of the file based roots, the VS requires some directories and files to exist for every used language.

- A **ID.lrs** file and a **ID.prs** file. E.g.: “01.lrs” and “01.prs”
- A directory that contains all pre-recorded prompts used for playback of this language. This directory is a base directory followed by the ID (E.g.: “C:\TOPCALL\Shared\Voicelib\TCVMail\01”).

Normally the PRS file is located under “C:\TOPCALL\Shared\VoiceLib\Sys”. And the LRS file is located under “C:\TOPCALL\Shared\VoiceLib**Plugin-Name**” (E.g.: “C:\TOPCALL\Shared\Voicelib\TCVMail”)

These paths are configurable.

9.3.2.3.2 The XML Way

The xml standard, and especially the VoiceXML standard, specifies a way to define the language of a text. This is done by placing the “xml:lang” attribute within a tag. E.g.:

```
<prompt xml:lang="en-UK"> Hello World </prompt>
```

The prompt “Hello World” has to be spoken by a British English TTS speaker.

In the following example a German speaker is requested.

```
<prompt xml:lang="de"> Hallo Welt </prompt>
```

The tokens “en-UK” or “de” are standardized by the IETF. You can find the standard here:

<http://www.ietf.org/rfc/rfc3066.txt>

To allow a mapping between “en-UK” and a specific TTS speaker the TTSConf tool is required (Figure 65 – TTSConf language).

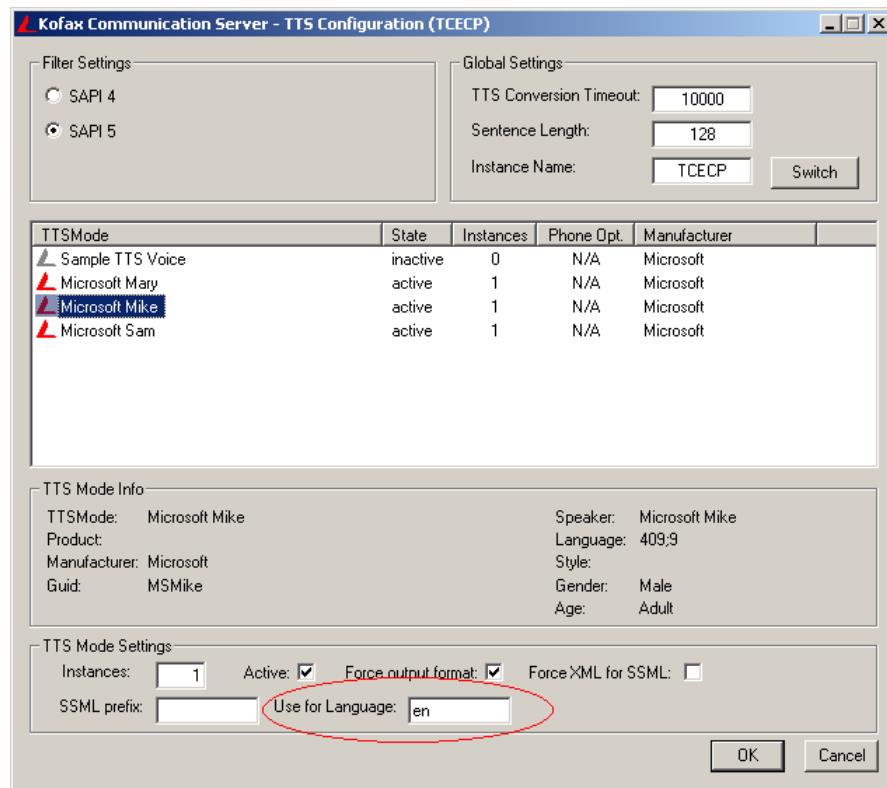


Figure 65 – TTSConf language

To have “Microsoft Sam” used as speaker for “en-UK”, activate “Microsoft Sam” and change the value next to “Use for Language” from “en” to “en-UK”.

If there are multiple TTS speakers configured for the same language token, automatically the first one is taken.

9.3.2.3.3 Related Topics

[TCSI Codepage Conversion \(VL\)](#)

[TCSI Codepage Conversion \(VA\)](#)

9.3.2.3.4 Configuration

To change the used default language, set the registry key “TCRT\DefaultLanguage” [REG_SZ] to the new language identifier.

To change the directories where the VS looks for the LRS and PRS file and the pre-recorded prompts edit the registry key “TCRT\ResDefaultPath” [REG_SZ]. This key holds a “;” (semi-colon) separated list of directories to use. If this key is empty, it will defaults to:

“C:\Topcall\Shared\VoiceLib\Sys;C:\Topcall\Shared\VoiceLib\TCVMail”

9.3.2.4 Tracing

TCRT offers a wide range of possible trace settings. Every engine has its own key and its own trace settings. In most cases a trace level of “3” is sufficient to detect the primary source of the problems.

9.3.2.4.1 Configuration

A list (Table 8 – TCRT Tracing) of keys and which trace they control. All keys are of type “DWORD” with the maximum value of “ff”.

Registry key	Meaning
TCRT\EngineTraceLevel	Controls the tracing of the tceengine. Most probably you will not need this registry key.
TCRT\HostTraceLevel	Primarily used to trace the communication between the engines.
TCRT\SndTraceLevel	Controls the tracing of the sound engine. In most cases this is not of interest because the VS uses a telephone engine instead of the sound engine for playback.
TCRT\ResTraceLevel	Used by the resource engine. The resource engine can do quite heavy tracing and the maximum trace level “ffff”. Do not use this heavy tracing in productive environment because it will significantly slow down the VS.
TCRT\TelTraceLevel	Controls the tracing of the LS1 engine.
TCRT\RemTraceLevel	Controls the tracing of the remote engine. This is useful if there are network problems.

Table 8 – TCRT Tracing

9.3.2.5 Location of the TCRT Engines

It might happen that TCRT engines for the VS are not located under “C:\TCOSS\System\EEngines”. It is possible to change the path where TCRT looks for the engines.

9.3.2.5.1 Configuration

The registry key “TCRT\EnginePath” [REG_SZ] holds the path to the folder where TCRT looks for loadable engines. It defaults to “C:\TCOSS\System\EEngines”.

9.3.2.6 Playback of Remote Prompts

The VS cannot only play prompts located on the VS itself or text via TTS, it is also capable of playing prompts from a remote server. The VS can access this server via http, https, ftp and may more protocols. The requested prompt is downloaded to the VS and stored in a cache for later re-use.

9.3.2.6.1 Prerequisites and Additional Installation

The file “tcDownloadFile.dll” and the files of the libwww must be within the DLL search path.

[Audio Caching](#) must be enabled.

9.3.2.6.2 Configuration

The tracing for this feature can be activated by the registry key “TCDownloadFile\tracelevel” [DWORD]. Set the key to “1” to enable the tracing. Set it to “0” to disable the tracing again. Note: This will only provide useful information if the debug version of the libwww is used.

TCRT will only download file which can be stored in the [Audio Cache](#). Therefore, both features share the registry key “TCRT\ResCacheProtocols” [REG_SZ]. This key holds a semi-colon separated list of protocols TCRT can download. This list must at least always hold the protocol “file”.

Valid protocols are:

Protocol	Tested
File	Yes
http	Yes
https	Yes
ftp	No
dav (webdav)	No
telnet	No
Gopher	No

Table 9 – Valid Transport Protocols for Prompt Playback

Per default only “file” and “http” are used.

The registry key “TCRT\ResCacheDownloadTMO” [DWORD] is the time in msec how long TCRT should try to download a file. The default is 100 msec. Keep this value short. Large values may lead to annoying pauses during playback when a file is not available.

9.3.2.7 Audio Caching

In previous versions the VS would convert every piece of text over and over again, even if the text did not change. The same thing would happen if the VS had to convert a audio file from one codec to a different codec or if the VS had to download a prompt from a web server (see [Playback Of Remote Prompts](#)). This is unnecessary and costs a lot of time/performance.

Now it is possible to tell the VS to cache the results of a conversion or a download.

Because a cache is not a permanent store and certain files must expire after a given time, the cache actually is two layered. (See Figure 66 – Audio Cache Layers).

This design comes with the issue that a file must not be deleted while someone is using it. As long as a file is used, it will stay in layer 1. After a timeout the file will be dropped to layer 2 but will still be useable if it is currently in use. At this point a request for the file will result in a “not found”. Again after a timeout, the file will finally be deleted.

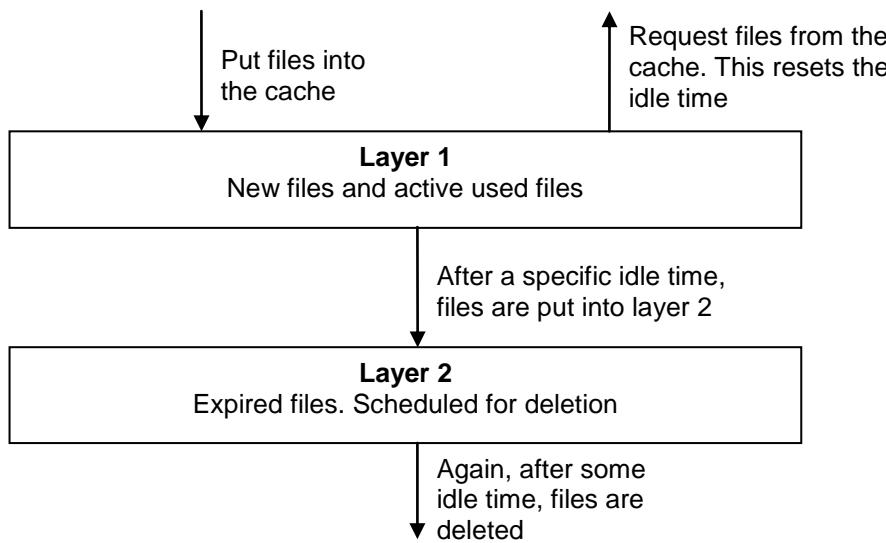


Figure 66 – Audio Cache Layers

But please note that under certain circumstances it might be preferable to disable the caching.

9.3.2.7.1 Related Topics

[Playback Of Remote Prompts](#)

9.3.2.7.2 Configuration

To enable or disable the audio cache set the registry key “TCRT\ResCacheEnabled” [DWORD] to “1” or “0”. It is “1” by default.

The cache will always cache the output of a TTS conversion but can be told not to cache the output of an audio codec conversion. This feature is called the ACM-Cache and can be enabled by setting the registry key “TCRT\ResCacheACM” [DWORD] to “1”. Setting the key to “0” disables this feature. The default is “1”

The registry key “TCRT\ResCacheDefTTL” [DWORD] specifies in seconds how long an unused file is kept in layer 1. The value defaults to “900” (15 min)

The registry key “TCRT\ResCacheExpiredTTL” [DWORD] specifies in seconds how long a file is kept in layer 2 before it is deleted. The value defaults to “1800” (30 min).

With the registry key “TCRT\ResCacheInterval” [DWORD] it is possible to tell TCRT how often to look for expired entries in both layers. The value is in seconds and defaults to “1800” (30 min)

The directory where TCRT has the cache can be configured with the registry key “TCRT\ResCacheDir” [REG_SZ]. It defaults to “C:\TCOSS\System\Cache\tce_res”. TCRT will create a subdirectory with the name of the VS instance it is running under. E.g.: “C:\TCOSS\System\Cache\tce_res\TCECP”

With the registry key “TCRT\ResCacheMaxSize” [DWORD] the maximum size of the cache on the HDD can be defined. The unit is MB and the default value is 1024MB (1GB).

Note: The maximum cache size is only applied to the first layer of the cache. This is due the fact that temporary files are stored and managed in the second layer of the audio cache. These temporary files could fill up the cache and degrade the performance behavior to a factor where it is worse than without cache.

9.3.2.8 SSML Support

SSML is the Speech Synthesizes Markup Language. Basically it is a XML dialect for prompt playback. It allows the user to change certain attributes (like TTS speed) during playback. Another feature is the possibility to specify alternatives if a prompt could not be loaded.

TCRT implements the part of the SSML specification regarding the playback of prompts. This is done because most TTS engines understand SSML but only the text and not the file part. To solve this problem, TCRT parses the SSML document and extracts all parts where the playback of a file is needed. The resulting text only SSML document is then passed to the TTS engine. This leads to the result that depending on the TTS engine more or less features of SSML specification are available.

9.3.2.8.1 Prerequisites and Additional Installation

The file "libexpat.dll" must be within a DLL search path.

SAPI 5. It is possible that a SAPI 4 TTS also recognizes SSML but very unlikely.

TTSConf

9.3.2.8.2 Related Topics

[TTS](#)

9.3.2.8.3 Configuration

No configuration in the registry is necessary but it may be necessary to work with TTSConf

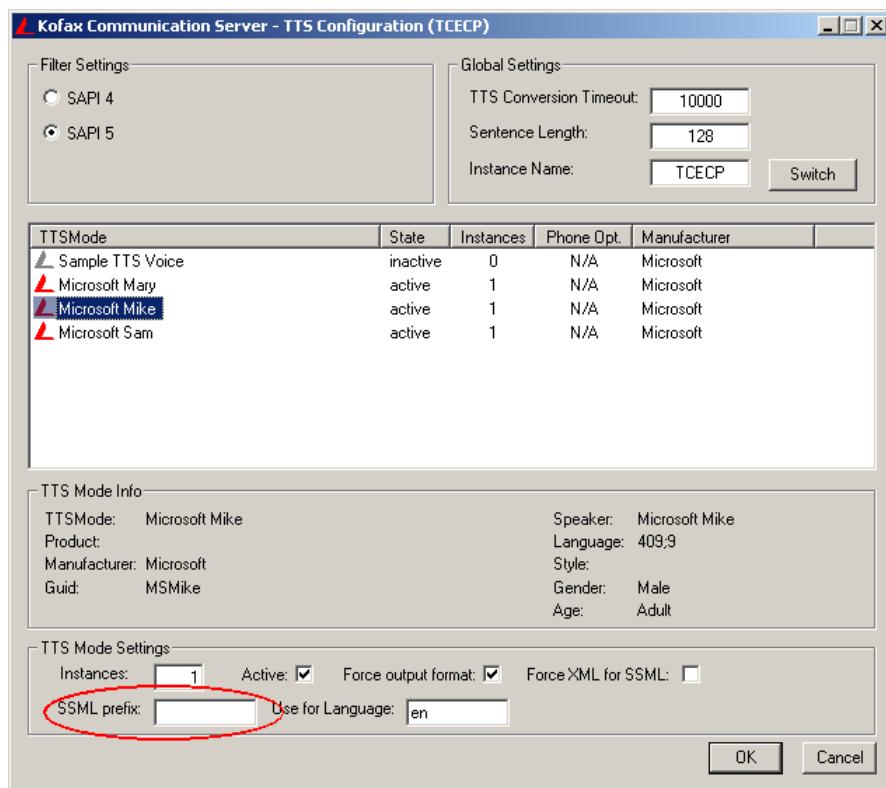


Figure 67 – SSML Prefix with TTSCConf

Some TTS engines require a special prefix to be added to the text when the text is SSML. Insert this prefix into the field labeled “SSML prefix” in TTSCConf (Figure 67 – SSML Prefix with TTSCConf)

To know what the correct prefix is, please refer to the manual of the TTS engine.

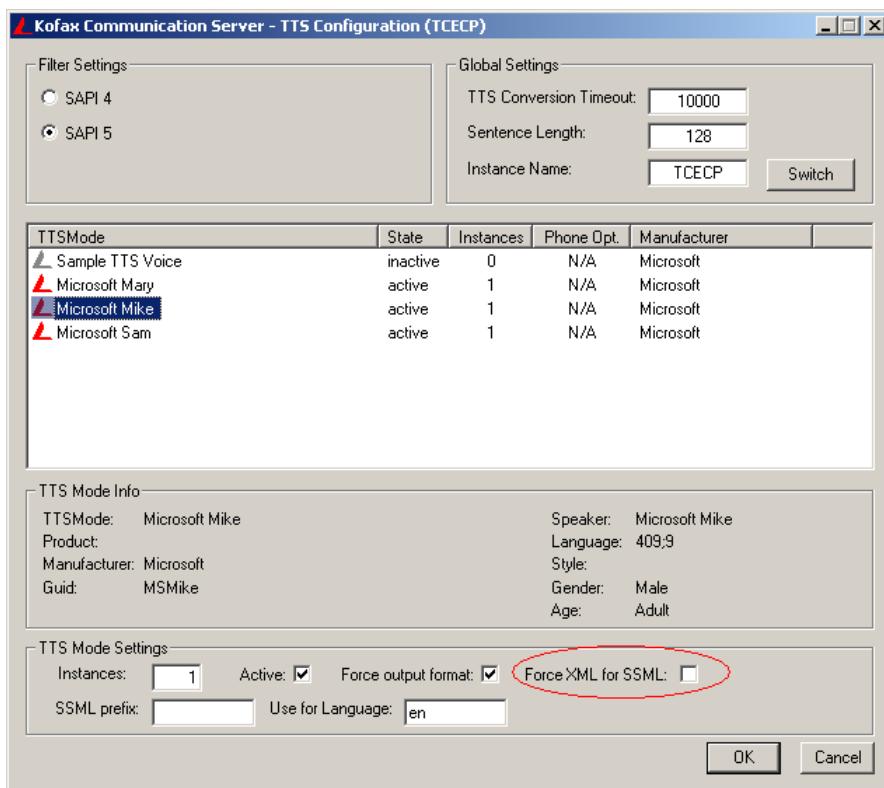


Figure 68 – Force XML for SSML

Some TTS engines will not recognize XML if it has an SSML prefix. To enable the TTS engine to treat SSML text as XML, check the “Force XML for SSML” checkbox (Figure 68 – Force XML for SSML)

9.3.2.9 System-Wide Default TTS Speaker

TCRT has some mechanisms to prevent the situation where a user is not able to hear a message because the TTS speaker is not configured correctly.

First it is possible to define a default speaker for each language. If this is not done, TCRT will take the first available TTS speaker and uses it for playback.

9.3.2.9.1 Related Topics

[Speech profiles](#)

[TTS](#)

[Email Reader Support](#)

[Languages](#)

9.3.2.9.2 Configuration

The default speaker for each language can be configured with the registry keys “TCRT\TTSSpeakerID” [REG_SZ] where “ID” is the language ID (see [Languages](#)). The value of the key must be a valid GUID of an active TTS engine. The GUID is displayed in TTSSConf. Unfortunately they look different for SAPI 4 and SAPI 5.

Here is an example for SAPI 4.

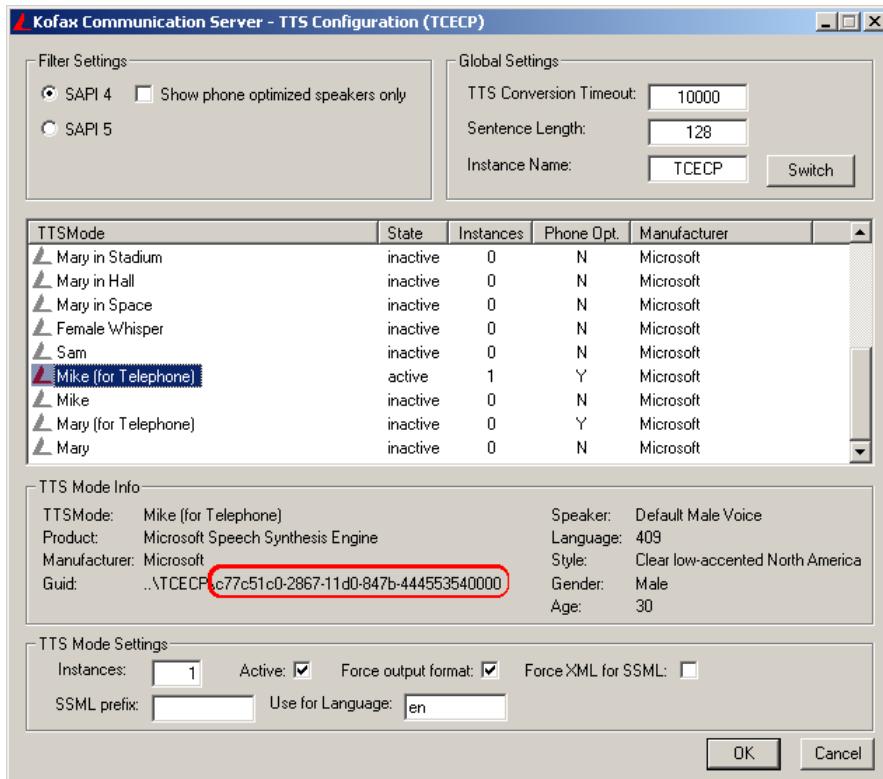


Figure 69 – SAPI 4 GUID

Note: The GUID is only the part after “..\TCECP\”

Here is the example for SAPI 5:

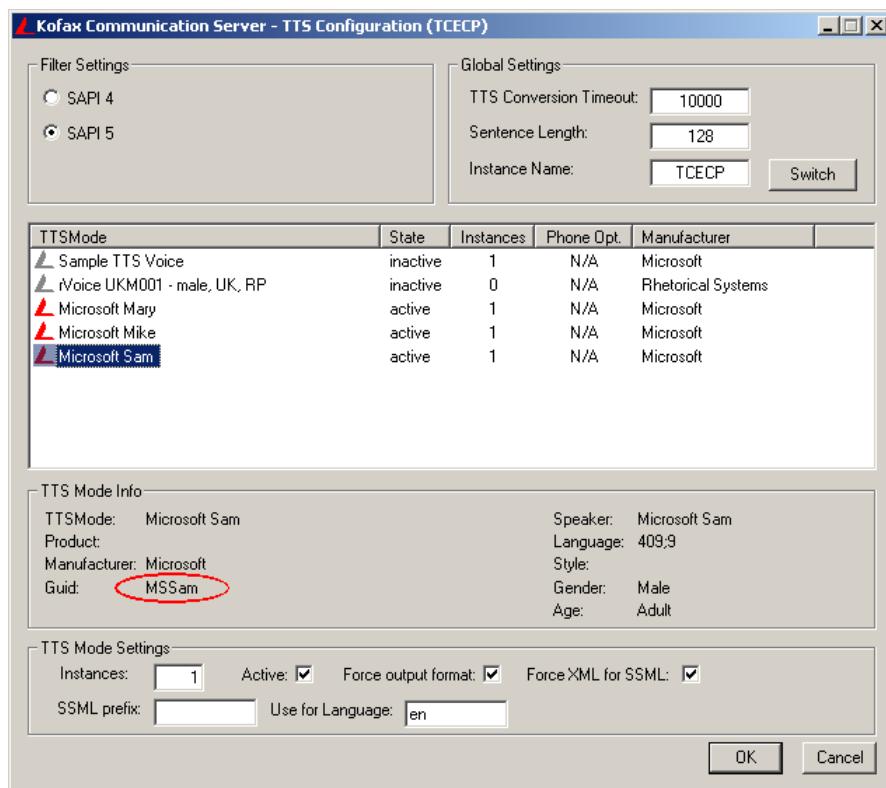


Figure 70 – SAPI 5 GUID

9.3.2.10 Voice Server Groups and Remote Connections

Each instance of a VS can belong to one or more voice server groups and each instance of a VS can expose multiple resources. TCRT automatically exposes all resources offered by the available engines.

This would be:

- “remote” for the remote engine.
- “sound” for the sound engine
- “telephone” for the telephone engine
- “resource” for the resource engine

But it is also possible to add so called “fake” resources. These fake resources are used to route connections to the correct VS and to the responsible plug-in.

9.3.2.10.1 Related Topics

[Voice Server Groups and Remote Connections](#)

[TRS – Transparent Routing Script](#)

9.3.2.10.2 Configuration

The group membership can be configured with the registry key “TCRT\RemGroupmembership” which can be:

- of type [REG_SZ] holding a list of semicolon ";" separated group names
- of type [REG_MULTI_SZ] where each line is a group name

To add fake resources use the registry key "TCRT\RemFakeResources" which can be:

- of type [REG_SZ] holding a list of semicolon ";" separated resource names
- of type [REG_MULTI_SZ] where each line is a resource name

When installing a VS the first time into an existing VSG, please enter the name of an existing VS of this VSG into the registry key "RemoteEngine\WellknownServers" [REG_SZ]

9.3.3 TC/VoiceAccess

9.3.3.1 The Attendant

When active, every time a caller wants to leave a message to a mailbox he will get a list of possibilities.

Currently a fully activated attendant would play: "To leave a message press 1. To call a different person press 8. To leave a fax press 9. To call the operator press 0".

Depending on activated features like [USN](#), [Alternative telephone number and representative](#) or [DialbyName \[DBNI\]](#) the dialog may vary.

The default Voice Attendant menu contains menu item 1 to "leave a message" and 0 to "call an operator". If the caller uses menu 1 "to leave a message", the VA will prompt the caller to leave a message after the tone. If the caller chooses "Contact the operator", the call will be transferred to the operator. See [Alternative operator](#) for additional information

For details about how VA establishes the connection see [Call Transfer and Loop Detection](#)

For details about possible loop detection, see [Call Transfer and Loop Detection](#)

Note: Since version 3.08.01, the user can enable the whole or parts of the attendant via TC/Web (version 4.08.04)

9.3.3.1.1 Related Topics

[Alternative operator](#)

[User Depending Languages](#)

[Call Transfer and Loop Detection](#)

[DialbyName \[DBNI\]](#)

[USN](#)

9.3.3.1.2 Configuration

When the attendant has not been activated during the setup, set the registry key "TCVMAIL\UseAttendant" [DWORD] to 1.

The attendant will not be activated if following conditions are met:

- If the called number is equal to the “TCVMAIL\ExceptionNumber”. This is useful for the operator because if the operator cannot answer the call, no attendant will be started and no loop can begin. Simply set the “TCVMAIL\ExceptionNumber” to the same value like the “TCVMAIL\TransferNumber”
- Called person has set his [Alternative operator](#) to his own voice extension. E.g.: The user has the extension 123 and his alternative operator has also the 123 then a caller will not hear the attendant.

There is one special case where the attendant is active but not played: If the key “TCVMAIL\PlayLeaveMsg” is disabled (0) and the user has recorded a personal greeting. In this case, the attendant prompts will not be played. This means that the user has to mention the options of the attendant during his personal greeting.

To change the song which is played while calling the operator, replace the files song1.wav in every directory in “C:\TOPCALL\shared\voicelib\tcvmail\”. The file has to be 8bit, 8kHz mono or a supported [Audio formats](#)

The administrator can disable the attendant individually for each user by adding the value 8 to the “flags” field of the currently active state in the user’s profile.

E.g.: The user’s current active state is “STATE4” and the flags are already set to 1, the new value must be $8+1 = 9$

```
tcusertool login:user;pwd filter:ts_user_id;user cmd:change /
set:SET_MAP.TOPCALL.TCVMAIL.STATE4.FLAGS;9;int
```

9.3.3.2 Customized Greetings and System Prompts

VA offers great customization possibilities regarding the welcome prompts and greetings.

The user has the ability to control if his phone extension, his name or his personal welcome greeting should be played.

The administrator has the possibility to turn on or off the system prompts and to specify a default language.

Note: This chapter will not describe how to customize the settings for each user; for information on this topic please see the VA user manual [\[8\]](#).

The settings made by the administrator have influence on the [Attendant](#), [Different languages for different access numbers](#) and on [User depending languages](#).

In case the administrator changes one of the following configurations, it is advisable to inform the users because they have to change their greetings according to the new settings.

9.3.3.2.1 Related Topics

[Different languages for different access numbers](#)

[User depending languages](#)

[The Attendant](#)

9.3.3.2.2 Configuration

Basically there are only two registry keys to configure the behavior of the welcome prompts: “TCVMAIL\PlayLeaveMsg” and “TCVMAIL\PlaySysMsg”.

"TCVMAIL\PlayLeaveMsg" can be set to 1 which activates or 0 which deactivates the prompt "Please leave a message after the tone". This prompt will be played after the user's personal greeting. If a user does not have a personal greeting recorded, "TCVMAIL\PlayLeaveMsg" has no influence.

"TCVMAIL\PlaySysMsg" can be set to 1 which activates or 0 which deactivates the prompt "Currently you cannot leave a message. Good bye". This prompt will only be played when a mailbox is disabled.

Additionally the key "TCVMAIL\AllGreetingsAreEqual" defines if VA should play different prompts for internal and external callers ("0" default) or if the external prompt is always used (set the key to "1").

Here is an exact table how it works:

PlayLeaveMsg	PlaySysMsg	Disabled Mailbox	Recorded Prompt	Play attendant	After pressing "1" you hear:
1	1	yes	played	played	"Currently you cannot..."
1	1	no	played	played	"Please leave a message ..."
0	1	yes	played	played	„Currently you cannot..."
0	1	no	played		
1	0	yes	played	played	„Goodbye"
1	0	no	played	played	„Please leave a message ..."

Table 10 – Customized Greetings and System Prompts

9.3.3.3 User Depending Languages

If VA is installed as an international voice system, special requirements regarding the prompts' language may rise. One point is that user may demand another language for the attendant. VA offers the possibility to configure the language of the attendant for each user separately.

9.3.3.3.1 Prerequisites and Additional Installation

An activated [Attendant](#)

9.3.3.3.2 Related Topics

[Different languages for different access numbers](#)

9.3.3.3.3 Configuration

To activate the feature, set the registry key "TCVMAIL\UserNotSysLanguage" to 1.

Afterwards, VA will use the language defined for each user instead of the default language.

If the user sets the language to a language that is not supported by the installed VA version, the default language will be used. This default language can be configured with the key "TCVMAIL\DefaultLanguage". The value can be a valid TC language ID.

The language of a user can be configured in TCfW.

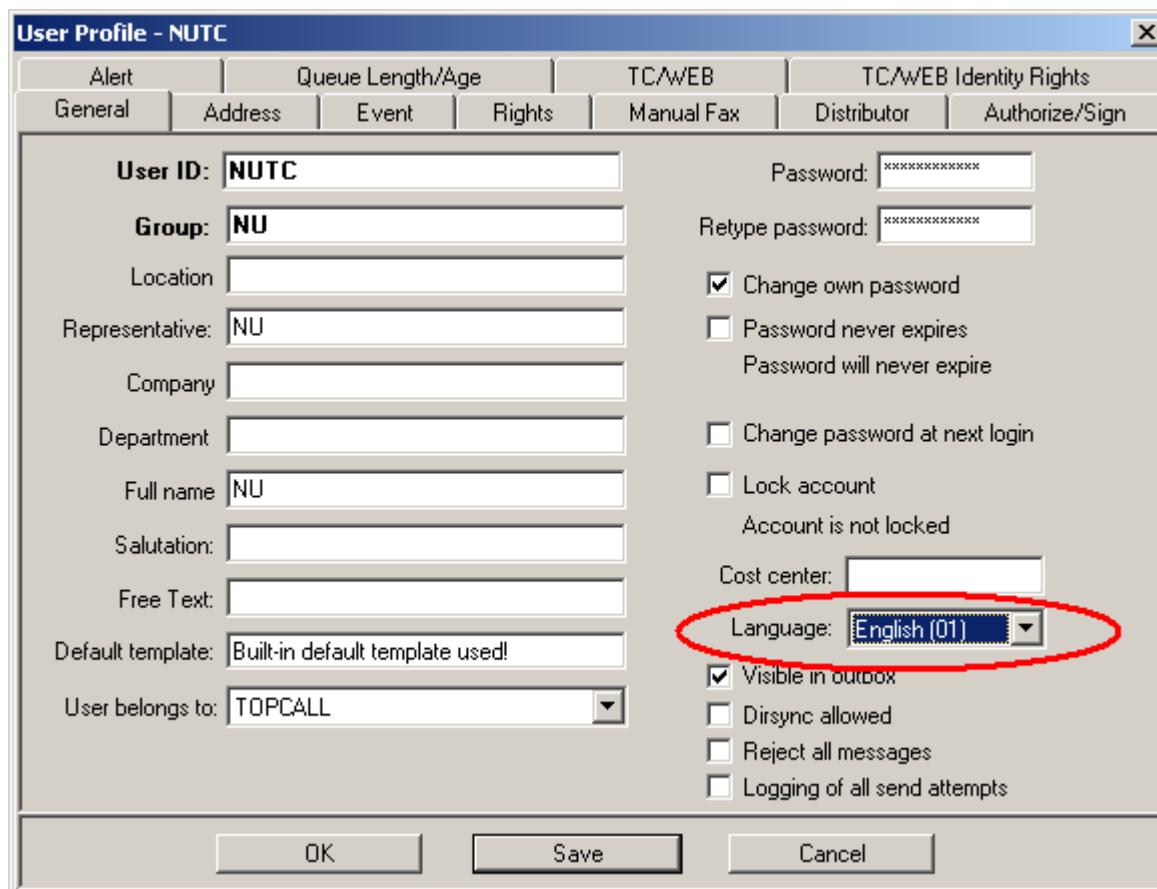


Figure 71 – Choose a Language in TCfW

9.3.3.4 Different Languages for Different Access Numbers

VA allows the usage of different languages for different VA numbers.

9.3.3.4.1 Related Topics

[User Depending Languages](#)

9.3.3.4.2 Configuration

During setup it is possible to enter the access numbers or they can also be entered directly into the registry ("TCVMAIL\AccessNumbers").

The standard and original form was to enter only the different access numbers separated by a ",". E.g.: 97,98,99

To specify a language, add a "/" followed by the TC language ID to the access number. If there is no language defined, the default language will be used. E.g.: 97/01,98/02,99

This default language can be configured in "TCRT\DefaultLanguage" The value can be a valid TC language ID.

Please see [TCRT - Languages](#) for details about the default language.

9.3.3.5 Alternative Operator

The task of an operator is to help callers who have reached a mailbox, and do not want to leave a message. The operator may connect this person to another extension or may give information about the called person. In big companies there are many operators and most likely, different operators for different user groups.

E.g.: In a company with many departments, each department has its own operator which is responsible for providing information about telephone numbers, status like “out of office” and other useful things to callers. If a caller tries to reach a person who cannot answer the call, the caller will hear a menu: „To leave a message, press 1, to contact the operator, press 0”. Now, the caller presses 0 and is connected to the operator of the called person’s department.

Sometimes it may happen that even this operator cannot answer the call, so the caller will hear again “To leave a message press 1, to contact the operator press 0”.

To prevent infinity loops, the operator disables his attendant.

This feature is also part of [The Attendant](#) and uses the same settings as described in [Call Transfer and Loop Detection](#)

9.3.3.5.1 Prerequisites and Additional Installation

A service “VOICE_O” is needed. This service is of type Free

9.3.3.5.2 Related Topics

[Call Transfer and Loop Detection](#)

[The Attendant](#)

9.3.3.5.3 Configuration

The alternative operator can be configured for each user separately as service “VOICE_O”. If VA finds such a service, the number of this service will be used when a caller wants to contact the operator.

If VA cannot find a service “VOICE_O”, it will use the number stored in the registry key “TCVMAIL\Transfernumber” instead.

You can customize the name of the service used for the alternative operator with the registry key “TCVMAIL\Service_Operator”.

9.3.3.6 CallSender

Call Sender allows you to call back the sender of the message while you are listening to the message by pressing “5” on the telephone.

After the user pressed “5”, VA will transfer the call to the originator of the currently heard message.

For details about how VA establishes the connection see [Call Transfer and Loop Detection](#)

For details about possible loop detection, see [Call Transfer and Loop Detection](#)

9.3.3.6.1 Prerequisites and Additional Installation

[Call Transfer and Loop Detection](#)

9.3.3.6.2 Related Topics

[Alternative Operator](#)

[The Attendant](#)

9.3.3.6.3 Configuration

During installation it is possible to specify the prefixes for external numbers and internal numbers. This can also be configured with the registry keys:

- “TCVMAIL\Callsenderprefix_External”
- “TCVMAIL\Callsenderprefix_Internal”
- “TCVMAIL\InternalNrMaxLength”

The key “TCVMAIL\Callsenderprefix_External” will be used, if VA decides that the originator called from external.

The key “TCVMAIL\Callsenderprefix_Internal” will be used, if VA decides that the originator called from internal.

“TCVMAIL\InternalNrMaxLength” is the maximum length of an internal telephone number. In the case VA does not find an inactive voice address of the originator in the TC address book, the length of the caller's telephone number shows whether the call is internal or external.

Depending on the scenario and the used telephone engine, additional configuration is needed:

How VoiceAccess decides what prefix should be used:

- Can VoiceAccess find an inactive voice address for the originator in the TC- address book? -> Internal Prefix
- Is the length of the telephone number of the originator <= the value in “TCVMAIL\InternalNrMaxLength”? -> Internal Prefix
- In the case both previous steps are negative, it is an external number! -> External Prefix

Additionally VA can be configured to use the Callers ID when establishing the connection to the sender. This means that the sender will see the telephone number of the caller on the display of his telephone instead of the number of VA.

To enable this, the key “TCVMAIL\CallSenderPostCallerID” has to be set to 1.

9.3.3.7 Email Reader Support

The idea of a unified mailbox is to access all forms of communication through one single inbox. This includes voice, fax, email and SMS. Where voice messages are audio messages and there is no problem to play them via telephone, all other forms contain written information. This information must be converted so that a user can listen to it. The common term for such a conversion technique is TTS (Text To Speech). VA uses Microsoft's SAPI 4.0 or SAPI 5.X to offer the user the freedom of choice which TTS engine should convert the emails to speech. This is essential because no TTS engine supports all the languages a user may need. A TTS engine may support different languages and a number of speakers for each language. A TTS speaker defines a specific type of voice. In most cases a speaker has a name like "Roger" and defines that this speaker is an English male voice.

Because the sympathy for a speaker is highly individual, VA offers [Speech profiles](#). This allows each user to tune his favorite speaker individually.

When the email reader support is activated, VA allows the user to filter his mailbox. This filtering is limited to "only voice mails" or "all messages in the inbox". Per default this filter is set to allow only voice mails. If the email reader support is deactivated, this filter will be set to "only voice mails".

9.3.3.7.1 Prerequisites and Additional Installation

SAPI 4.0 or SAPI 5.X

A supported TTS Engine

The TTS speakers must be enabled and configured via TTSConf.

9.3.3.7.2 Related Topics

[Supported TTS Engines](#)

[Speech profiles](#)

[TTS Engine and Speaker Configuration](#)

[VL - Disable TTS](#)

9.3.3.7.3 Configuration

After the installation, do not forget to configure the TTS engines with TTSConf.

During installation of the KCS, check the option "enable email reader support..."

This option can be set manually with the registry key "TCRT\ResEnableTTS" [DWORD]. If set to 1, the email reader support will be enabled. 0 disables the email reader support.

Be aware that the VL does not check this key. How to tell the VL to stop using the TTS is described in [Disable TTS](#).

9.3.3.8 Minimum Recording Length

By using the minimum recording length, all messages shorter than the defined length will be discarded and not sent to the user mailbox.

This only applies if a caller leaves a message. When a user composes, forwards or replies to a message, the minimum recording length is not used.

9.3.3.8.1 Related Topics

[Maximum Recording Length](#)

9.3.3.8.2 Configuration

Set the key “TCVMAIL\MsgLengthMin” to the minimum length of a recorded message. The length is specified in msec.

E.g.: If set to 500, every message shorter than 500msec will be discarded.

9.3.3.9 Maximum Recording Length

By using the Maximum Recording length, recording will stop and VA will disconnect the call if the specified duration is reached.

9.3.3.9.1 Prerequisites and Additional Installation

This feature has to be supported by the used telephone engine.

9.3.3.9.2 Related Topics

[Minimum Recording Length](#)

9.3.3.9.3 Configuration

Set the key “TMOLength” to the wanted length in sec. After this time period, VA will stop recording and hang up. The recorded message will be sent.

9.3.3.10 Send a Fax with Voice

This feature allows printing or sending fax messages, while the user is listening to messages in case an email or fax has arrived.

This feature can be used when forwarding a message. If the user forwards a message he will be asked to “enter list of mailbox numbers or fax numbers”.

9.3.3.10.1 Prerequisites and Additional Installation

The possibility to send faxes via TCOSS.

9.3.3.10.2 Related Topics

[Dial by Name and Distribution Lists](#)

9.3.3.10.3 Configuration

The fax functionality can be configured during setup or within the registry.

The field “Prefix of fax numbers” is represented as “TCVMAIL\PrefixFaxNr” in the registry and is also referred as fax prefix.

The field “Min. length of Fax numbers” is represented as “TCVMAIL\MinFaxNrLength” in the registry.

The detection if the entered number is a fax depends on whether the feature [Dial by Name and Distribution Lists](#) is enabled or not. If the feature has been disabled (default setting), VA will search for a valid fax number like described below.

1. Is the entered number **no** valid voice address? Proceed with step 2 otherwise it is no fax.
2. Has the entered number **no** prefix like it is defined in “TCVMAIL\PrefixFaxNr”? Proceed with step 3 otherwise it is a fax.
3. Is the length of the entered number at least as long as “TCVMAIL\MinFaxNrLength” says? It is a fax, otherwise it is no valid number at all.

If [Dial by Name and Distribution Lists](#) are enabled, the user will be asked if the entered number is a fax number.

9.3.3.11 Dial by Name and Distribution Lists

The DistList support requires DBN. Without DBN it would not be possible to address DistLists because they cannot have a voice address.

What is DBN

DBN enables the caller and user to use names of recipients instead of their voice address. This means that a user who composes a message does not need to enter the recipient's mailbox number anymore. He can simply enter the name of the recipient. In TCOSS either the full name or the short name is used for lookup.

DBN behaves different for callers and users.

For the caller:

To enable DBN for the caller, also the attendant has to be enabled. If a caller hears the attendants menu, a new option will be played (“To call a different person press 8”). After pressing 8, he can search for a user or recipient out of the TCOSS system address book by typing the user's name on the telephone as “vanity style”. The user or the recipient must have a voice address.

With DBN, the voice message menus for “compose” and “forward” are enhanced so that the user can address any recipient from the system address book. This includes DistLists and does not depend on a specific address type. It is still possible to address faxes.

DBN uses the norm E.161 Option A (05/95) “vanity style” for the matching of numbers to characters. For details about how this works please refer to [\[6\]](#)

What means support of DistLists

The DistList support enables the user to send a message to a DistList instead of a single recipient. VA will resolve this DistList and sends each recipient in this DistList the message. Only a user can use DistLists within his mailbox. A caller cannot address his message to a DistList.

VA will not validate recipients from a DistList. This means that the DistList is not limited to use a specific service like “VOICE” or “FAX”.

VA will resolve DistList recursively. Because VA resolves DistLists internally, it may take some time before a message is sent. In short: "Do not send large DistList with VA. It may lead to a session time out."

VA does not define any limits for DistLists, but it is possible that there are limits through other modules. See [\[9\]](#)

Note: The look up for recipients and DistLists is only done within the system address book.

Note: To establish a connection, DBN uses [Call Transfer and Loop Detection](#).

How DBN and DistList work in detail and can be configured for a caller or a user, see [\[6\]](#)

9.3.3.11.1 Prerequisites and Additional Installation

DBN support has to be activated in TCOSS. See [\[6\]](#)

9.3.3.11.2 Related Topics

[Call Transfer and Loop Detection](#)

[The Attendant](#)

9.3.3.11.3 Configuration

To enable DBN for the Caller, [The Attendant](#) has to be activated and the key "TCVMAIL\.DialbyName" has to be set to 1.

To enable DBN for the user it is not necessary to activate the attendant, it is sufficient to set the key "TCVMAIL\.DialbyName" to 1.

Because it is likely to get a lot of results when searching only for the first two characters of a name, it is possible to ask the user to enter more digits. Only if the user or caller enters more digits, DBN will search for a person's name. If the user or caller entered less digits only a mailbox and fax look up will be done. The key to do this is "TCVMAIL\DBN_MinVanityLength"

To specify when VA should play the full name of a user or the voice extension, the key "TCVMAIL\DBN_MinNameLength" has to be set to the minimum amount of characters a full name must have.

How VA determines what to play:

- 1.) Has the recipient recorded a name? If yes: Play the name. No: Step 2.
- 2.) Is the full name of the recipient longer or equal to "TCVMAIL\DBN_MinNameLength"? If yes: Play the full name with TTS. No: Step 3.
- 3.) If a voice extension was found, play it. If there was no voice extension, play "0000"

9.3.3.12 Unique Single Number

The idea behind USN is to provide only one number for telephony (voice) and fax. This offers cost reductions for the company in countries where you have to pay for each extra digit in your telephone number. E.g.: When using a 3 digit telephone extension where every user needs 2 extensions (one for fax and one for voice), it is necessary to buy a forth digit as soon as there are more than 500 users.

Because USN must be supported by the telephone system, please refer to [Restrictions](#) for known problems.

9.3.3.12.1 Prerequisites and Additional Installation

It is necessary to configure the fax and voice channels on TCOSS correct so that a line can handle fax and voice. See [\[5\]](#) for details

9.3.3.12.2 Configuration

To enable USN the attendant has to be active and the registry key "TCVMAIL\doUSN" is set to 1. Other values than 0 or 1 are currently not supported.

When enabled, the caller will have an additional option in the attendants menu ("To leave a fax press "9").

9.3.3.13 Immediate Response

With Immediate Response turned on, the user will hear "Please wait - Your mailbox will be opened" until VoiceAccess has collected all data from the user's mailbox.

9.3.3.13.1 Configuration

Set the key "TCVMAIL\ImmediateResponse" to > 0 to activate it.

9.3.3.14 Customized Reply/Forward Header

When forwarding a message or replying to a message, VA includes the header of the original message into the new one. It also adds a prefix to the subject.

It is possible to turn on/off the included header and it is possible to define the prefixes.

This is especially interesting when no TTS is installed because to read the header, VA would need a TTS.

An included header looks like:

```
From:  
TOPCALL, NUTC  
To: (TOPCALL, NUTC)  
    TOPCALL, NUTC  
    TOPCALL, NUMX  
    TOPCALL, NULN  
CC:  
Date:27.11.2003 11:08:40  
Subject:Voice mail from NU TOPCALL, 661 [3sec]
```

9.3.3.14.1 Configuration

When the key "TCVMAIL\IncludeFWD_REPL_Headers" is set to > 0 the header of the original message is included.

The key "TCVMAIL\ForwardPrefix" specifies the prefix added to the subject when the message is forwarded. E.g.: If "TCVMAIL\ForwardPrefix" holds "Fwd:", a message that is forwarded will have a subject like "Fwd: Look at this interesting subject".

The key “TCVMAIL\ReplyPrefix” specifies the prefix added to the subject when the message is replied. E.g.: If “TCVMAIL\ReplyPrefix” holds “Re:”, a message that is replied will have a subject like “Re: Fwd: Look at this interesting subject”.

9.3.3.15 Customizable Filename

VA attaches the recorded voice mail as “Message.wav” to a message. Comments from replies or from forwarded messages are attached as “Comment.wav”. This is language independent and can be changed.

9.3.3.15.1 Prerequisites and Additional Installation

During the installation only the file extension for received voice mails can be defined. (“wav” or “tcs”). This has no influence on the audio format. The difference is that when using “wav” the standard audio player will open the file. When using “tcs” the TC/Player will open the file.

The file extension is stored in the key “Extension”

9.3.3.15.2 Configuration

The key “TCVMAIL\FNameMessage” holds file name of the attachment that is created when a caller leaves a message. Only the file name without extension is stored in this key. The extension is stored in “Extension”.

The key “TCVMAIL\FNameComment” holds file name of the attachment that is created when a user replies or forwards a message with a comment. Like “TCVMAIL\FNameMessage” only the file name without extension is stored.

Both keys (“TCVMAIL\FNameMessage”, “TCVMAIL\FNameComment”) may not be longer than 8 characters. Otherwise they may be truncated.

9.3.3.16 Call Transfer and Loop Detection

Additionally to the concept already explained in Concepts and Architecture, VA supports a simple mechanism to prevent loops on calling the (same) operator extension again and again. This may occur if the caller has already selected to be connected to the operator but the operator is also busy or not responding and this call is therefore being forwarded to operator’s mailbox on the Kofax Communication Server.

In the case the operator’s mailbox has already been reached, VA offers the caller only the option to leave a message but never more to be connected to the operator.

The operator’s mailbox is the mailbox whose mailbox number (extension) equals to:

- Either the inactive VOICE_O address (personal attendant) within own user profile
- Or with the Exception number configured in the registry key “TCVMAIL\ExceptionNumber [REG_SZ]

9.3.3.16.1 Prerequisites and Additional Installation

For TCOSS and the telephone engine see: [\[5\]](#) and [\[7\]](#)

9.3.3.16.2 Related Topics

[Call Transfer and Loop Detection \(Concepts and Architecture\)](#)

9.3.3.16.3 Configuration

9.3.3.16.3.1 Call Transfer

For [The Attendant](#) the key “TCVMAIL\AttendantTransferMode” (default is “EXT”)

For [CallSender](#) the key “TCVMAIL\CallSenderTransferMode” (default is “INT”)

Possible values are “EXT” and “INT”.

“EXT” means: External call transfer.

“INT” means: Internal call transfer.

When using [CallSender](#), the called person will see the telephone number of VA server. If the key “CallSenderPostCallerID” is set to “1” (recommended), the called person will see the mailbox number of the caller and if set to “2” the called person will see the telephone number of the caller. “0” is the default where the called person only sees the number of the VA server.

For [The Attendant](#) the key “TCVMAIL\AttendantPostCallerID” has the same behavior as mentioned above for “TCVMAIL\CallSenderPostCallerID”

9.3.3.16.3.2 Loop Detection

For [The Attendant](#) the key “TCVMAIL\AttendantLoopDetection”

For [CallSender](#) the key “TCVMAIL\CallSenderLoopDetection”

Both keys will take as values “SBOX” or “PBOX” where “SBOX” is the default behavior.

“PBOX” means that when a loop occurs, it is redirected to the original mailbox (the quoted “User A”)

“SBOX” means that when a loop occurs, it is redirected to the mailbox of the user which was called at last (the quoted “User C”)

9.3.3.17 Mailbox Sort Order

With the mailbox sort order, it is possible to configure in which order VA should play the messages in the unread and read sections. In general, there are only 2 possibilities: That VA starts with the newest or with the oldest messages.

Note: When using TCOSS as mail system, the sort order uses 2 steps. At first it sorts after priority and then after date and time. E.g. Three Message A, B and C are created in this order (A first, then B, then C).

Message A and C have the priority normal and message B has the priority high. When sorted “normally”, at first B will be played, then A and at last C. When the sort order is reversed, at first C will be played then A and at least B.

9.3.3.17.1 Configuration

To configure the unread sections sort order, use the key: "TCVMAIL\NewestFirstForNew" which has the default "0". This means that at VA will play at first the oldest unread messages and then the newer unread messages. If the key is set to "1", VA will play at first the newest unread message and then the older unread messages.

To configure the read sections sort order, use the key: "TCVMAIL\NewestFirstForOld" which has the default "1". This means that at VA will play at first the newest read messages and then the older read messages. If the key is set to "0", VA will play at first the oldest read message and then the newer read messages.

9.3.3.18 Sort Mailbox

Even if it sounds like [Mailbox Sort Order](#), it is not the same and describes 2 different things.

Whereas the mailbox sort order sorts the messages after priority and date/time, Sort Mailbox categorizes the messages in 3 different types. These types are "voice mails", "faxes" and "emails".

If this feature is activated, a user who accesses the mailbox will hear "You have XXX new voice mails, YYY new faxes and ZZZ new emails". The messages are also played in this order. This means that at first the user will hear the new voice mails (sorted after the mailbox sort order for unread messages), then the faxes and then the emails. After the user listened to all the new messages VA will play "You have XXX already heard voice mails, YYY already heard faxes and ZZZ already heard emails".

9.3.3.18.1 Prerequisites and Additional Installation

It is necessary that the TTS (Email reader) is installed

9.3.3.18.2 Configuration

To enable this feature set "TCVMAIL\SortMailbox" to "1". To disable it, set the key to "0".

9.3.3.19 Greeting Scheduler and Greeting Group

A standard voice profile is the plain old voice profile where the greetings and the necessary information are stored within the user's user profile.

A Greeting Groups [GG] is always a link that points to a standard voice profile and only contains as little information as possible

GG should help the administrator to create and administer the voice users. The most visible advantages are:

- It saves a lot of disk space. Each user stores only a link to a voice profile of the SysUser instead of the whole voice profile.
- The Administrator can easily change the greeting which is played by default.
- It saves a lot of time because not each user has to record an own greeting.

A Greeting Scheduler [GS] allows a user to define when a specific voice profile should be played. Without GS a user can only activate one voice profile which will be played until the user activates another one. With GS the user can specify that the voice profile A is played from 8:00 to 12:00 and the voice profile B is played from 12:00 to 17:00.

By default, GS are stored in the user profile. This allows each user to edit his GS. Additionally the administrator can create a GG for the user which points to a GS of the SysUser.

For the caller there is no difference between a standard voice profile, a GG or a GS.

When using VA to administer his mailbox, a user can lock or unlock his mailbox if the currently active voice profile is a GG or a GS.

9.3.3.19.1 Prerequisites and Additional Installation

For configuration purposes the TC/Web or the TC/Usertool may be used. The TC/Usertool is installed with the KCS (7.59.00 or higher) when the menu-item TC/VoiceAccess was selected.

9.3.3.19.2 Configuration

No additional configuration on the server side is necessary.

For the user the configuration can be done via TC/Web. See [\[19\]](#) for details about the user interface.

A user may create GS and standard voice profiles but he is not allowed to create GG.

GG can be created only by the administrator with the TC/Usertool.

Example how to configure a GG:

We assume that the voice profile “STATE4” with the name “holiday” should be configured as GG and should point to the voice profile “STATE2” of the SysUser.

1.) Set the flags for the voice profile to 0x10000000 (decimal 268435456) and 0x20000000 (decimal 536870912) which means that the voice profile is a GG and the flags are located in the SysUsers profile. Type:

```
tcusertool login:user;pwd filter:ts_user_id;user cmd:change /  
set:SET_MAP.TOPCALL.TCVMAIL.STATE4.FLAGS;805306368;int
```

2.) Set the Link to the name of the voice profile of the SysUser (“STATE2”). Type:

```
tcusertool login:user;pwd filter:ts_user_id;user cmd:change /  
set:SET_MAP.TOPCALL.TCVMAIL.STATE4.Link;STATE2;str
```

For more information how to use the TC/Usertool, see [\[18\]](#)

9.3.3.19.3 Field Definitions

GS and GG are stored like every other voice profile in the user profiles SET_MAP area:

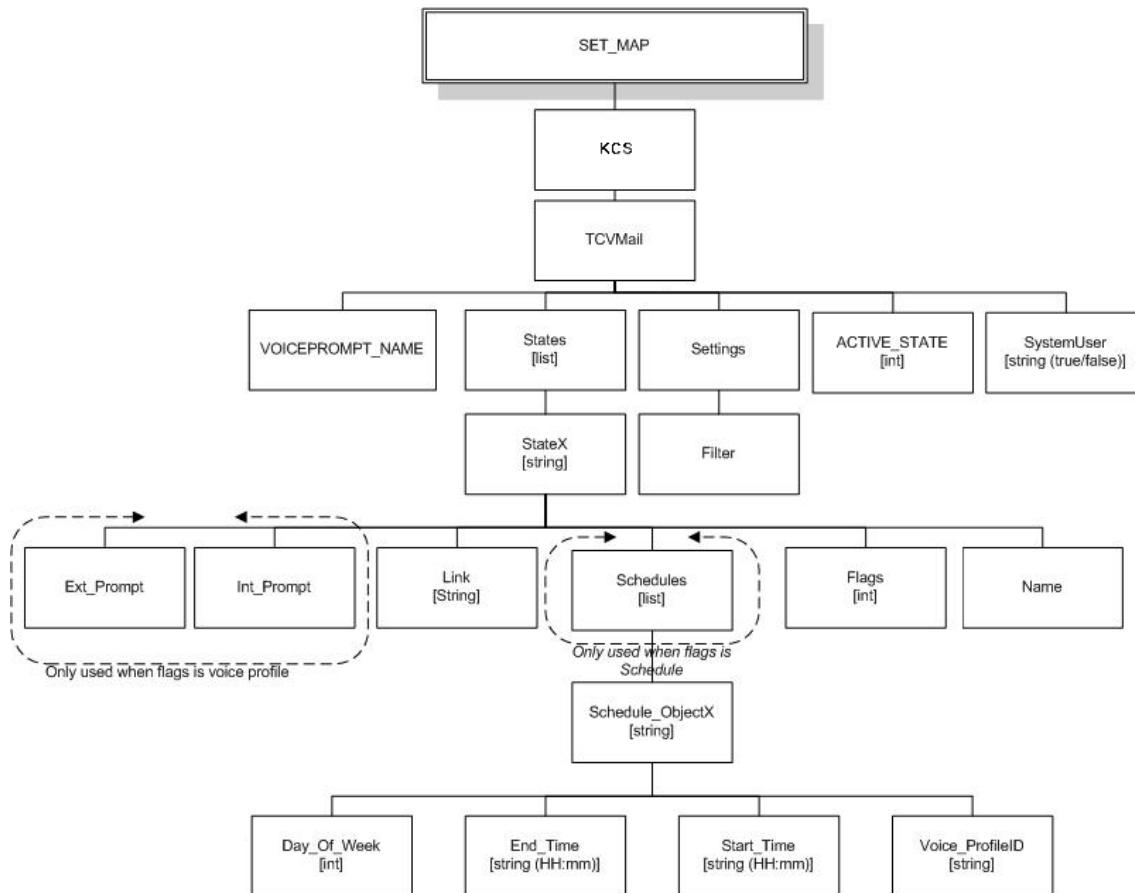


Figure 72 – TCSI – Tree for Greeting Groups and Scheduler

The field “ACTIVE_STATE” defines which state is active. It only stores an integer which is appended to the string “STATE”. The resulting string is the path to the active state.

E.g.: when the field “ACTIVE_STATE” holds the integer 4. The path to the active state is “SET_MAP.TOPCALL.TCVMail.States.STATE4”

Note: “StateX” is also referred as path name.

The field “Link” stores a path name. E.g.: “STATE4”. This path name points to the voice profile of the SysUser.

The field “Name” is the name of the voice profile that is displayed in TC/Web or TC/MC.

The field “Flags” defines the type and behavior of the state. Possible values are:

Value in hex (dec)	Description
0x1 (1)	There is also an internal greeting
0x2 (2)	It is not possible for external callers to leave a message
0x4 (4)	It is not possible for internal callers to leave a message (This is only relevant when the flag 0x1 is set)
0x10000000 (268435456)	Identifies a GG
0x20000000 (536870912)	Indicates that the flags of the SysUsers voice profile have to be used. (Only relevant when the flag 0x10000000 is set)
0x40000000 (1073741824)	Identifies a Scheduler

Table 11 – Greeting Flags

“Schedule_ObjectX” behaves like “StateX” and is also referred as schedule path. “Schedule_Objects” are simply numbered from 0 to X.

“Day_Of_Week” is a bit-field that specifies the days where a schedule object is active.

Possible values are:

Bit	7	6	5	4	3	2	1
Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday

Table 12 – Scheduler Flags

“End_Time” and “Start_Time” stores the End/Start-Time of the Schedule Object in the the 24 hours format (HH:mm)

“VoiceProfile_ID” is the path name of a voice profile of the same user. (Unlike “Link” that points to a voice profile of the SysUser)

9.3.3.19.4 Behavior

The following activity diagram shows how VoiceAccess gets a prompt or the flags:

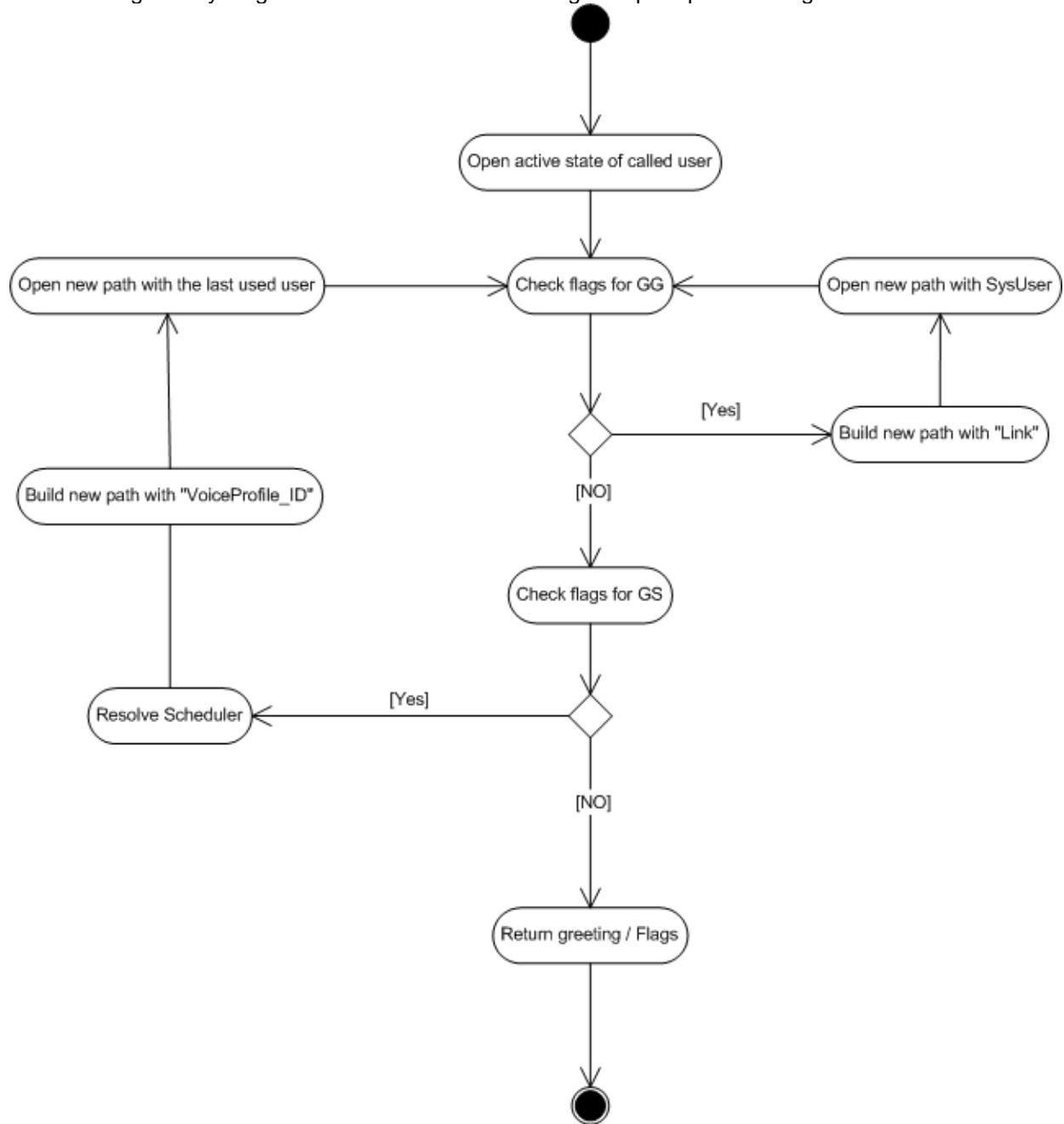


Figure 73 – VA Flowchart to Get a Greeting

To set a flag, VoiceAccess proceeds differently:

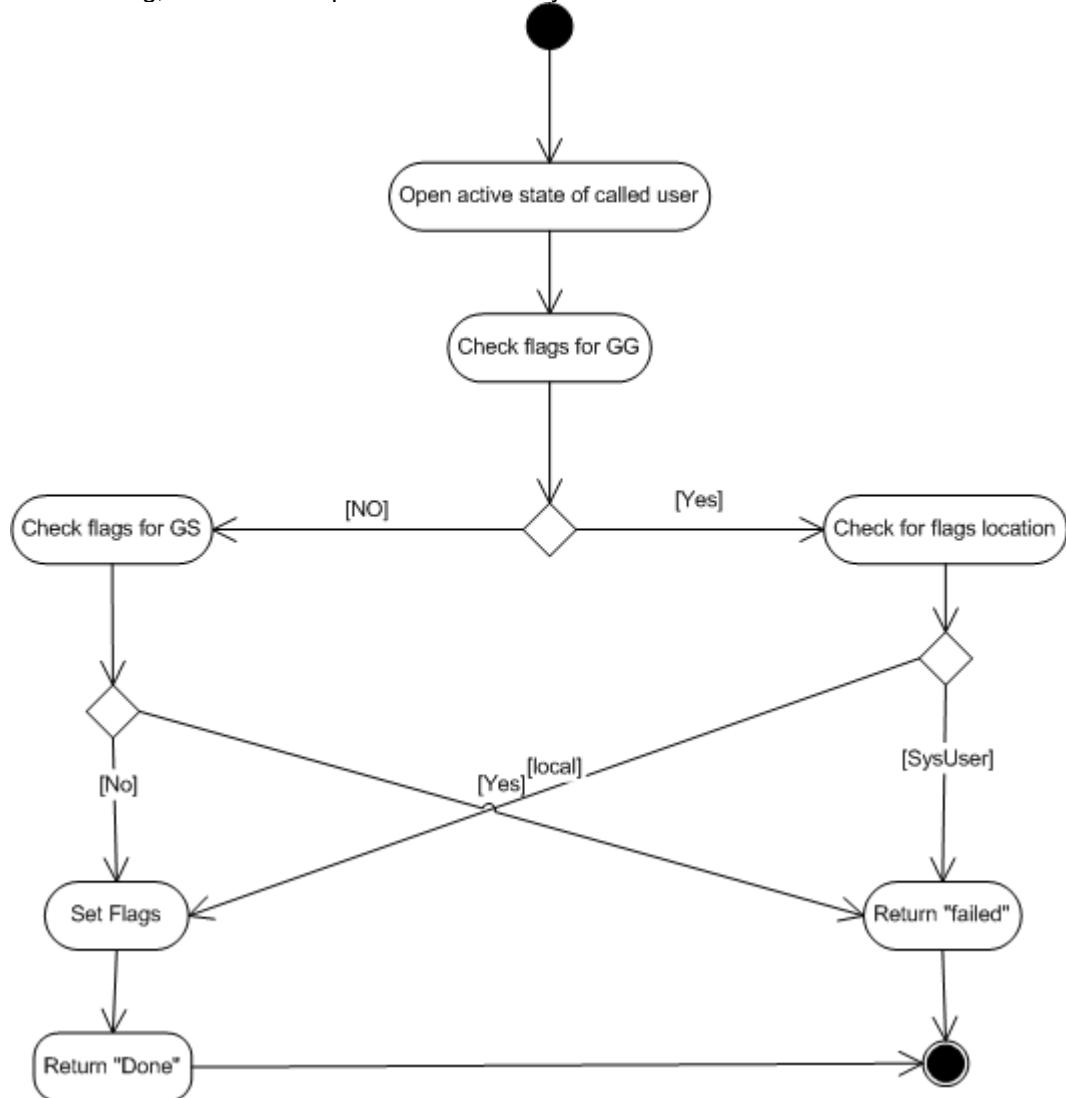


Figure 74 – VA Flowchart to Set a Greeting

This boils down to the following statements:

- A GG always points to a profile of the SysUser
- Nested GG are possible (Up to 255 times).
- Flags located in a profile of the SysUser cannot be changed from VoiceAccess or TC/Web
- A GG may point to a GS of the SysUser.
- A GS may point to a GG.
- GS and GG may be nested (Up to 255) times.
- The flags of a GS may not be changed via VoiceAccess.
- It is not possible to plan a specific day or week with GS (e.g.: 24.12.2005)

9.3.3.19.5 Examples

The following example can be used with the TC/Usertool. Copy it into a file, remove the comments and change the settings to fit your needs.

As greetings the files "local1.wav" and "local2.wav" are used.

At the command line type : **TCUserTool -f:yourfile.txt**

```
[login:]
yourserver;systemuser;systemuser
[cmd:]
change
[filter:]
ts user id;VoiceUser

[set:]
//This is a simple voice profile. It is local and uses local1.wav as greeting
SET_MAP.TOPCALL.TCVMail.STATESS.State1.Name;MyLocalGreeting1;str
SET_MAP.TOPCALL.TCVMail.STATESS.State1.Flags;0;int
SET_MAP.TOPCALL.TCVMail.STATESS.State1.EXT_PROMPT;.\local1.wav;att

//This is a GG which also uses the remote flags
SET_MAP.TOPCALL.TCVMail.STATESS.State2.Name;MyLinkGreeting;str
SET_MAP.TOPCALL.TCVMail.STATESS.State2.Flags;805306368;int
SET_MAP.TOPCALL.TCVMail.STATESS.State2.Link;State1;str

//This is a GG with local flags where the internal caller can leave messages but the external
can't
SET_MAP.TOPCALL.TCVMail.STATESS.State3.Name;MyComplexGreeting;str
SET_MAP.TOPCALL.TCVMail.STATESS.State3.Flags;268435459;int
SET_MAP.TOPCALL.TCVMail.STATESS.State3.Link;State2;str

//A voice profile where the mailbox is locked
SET_MAP.TOPCALL.TCVMail.STATESS.State4.Name;MyLocalGreeting2;str
SET_MAP.TOPCALL.TCVMail.STATESS.State4.Flags;2;int
SET_MAP.TOPCALL.TCVMail.STATESS.State4.EXT_PROMPT;.\local2.wav;att

//A GS that uses only local voice profiles.
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Name;MyLocalSchedule1;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Flags;1073741824;int
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object1.Days_Of_Week;124;int
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object1.Start_Time;8:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object1.End_Time;17:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object1.Voice_ProfileID;State1;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object2.Days_Of_Week;124;int
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object2.Start_Time;0:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object2.End_Time;8:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object2.Voice_ProfileID;State4;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object3.Days_Of_Week;124;int
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object3.Start_Time;17:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object3.End_Time;24:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State5.Schedules.Schedule_Object3.Voice_ProfileID;State4;str

//A GS that also uses GG
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Name;MyLocalSchedule2;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Flags;1073741824;int
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object1.Days_Of_Week;126;int
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object1.Start_Time;8:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object1.End_Time;15:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object1.Voice_ProfileID;State1;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object2.Days_Of_Week;124;int
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object2.Start_Time;2:30;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object2.End_Time;12:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object2.Voice_ProfileID;State4;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object3.Days_Of_Week;124;int
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object3.Start_Time;17:45;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object3.End_Time;24:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object3.Voice_ProfileID;State4;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object4.Days_Of_Week;127;int
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object4.Start_Time;5:00;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object4.End_Time;16:15;str
SET_MAP.TOPCALL.TCVMail.STATESS.State6.Schedules.Schedule_Object4.Voice_ProfileID;State2;str
```

```

//A GG that points to a GS in the SysUsers profile. To make this true. A GS for the SysUser has
to be created too.
SET MAP.TOPCALL.TCVMail.STATE.S.State7.Name;MyLinkSchedule1,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State7.Flags;268435456,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State7.Link;State3,str

//A GS where the greeting is only active at exactly 1:00 o'clock
SET_MAP.TOPCALL.TCVMail.STATE.S.State8.Name;MyEmptySchedule1,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State8.Flags;1073741824,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State8.Schedules.Schedule_Object1.Days Of Week;126,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State8.Schedules.Schedule_Object1.Start_Time;1:00,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State8.Schedules.Schedule_Object1.End_Time;1:00,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State8.Schedules.Schedule_Object1.Voice_ProfileID;State1,str

//A voice profile with nothing in it.
SET MAP.TOPCALL.TCVMail.STATE.S.State9.Name;MyEmptyGreeting1,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State9.Flags;0,int

```

The next example is a TC/Usertool configuration for the SysUser.

This example holds the counterparts to the previous example. It uses the files “remote1.wav”, “remote2.wav” and “remote3.wav” as greetings.

```

[login:]
yourserver;systemuser;systemuser
[cmd:]
change
[filter:]
ts_user_id;SysUser
[set:]
//Is linked by State1 and indirect by State6 from the voice user and state3 of the SysUser
SET_MAP.TOPCALL.TCVMail.STATE.S.State1.Name;TCVMAIL Greeting1,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State1.Flags;0,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State1.EXT_PROMPT;.\remote1.wav;att

//Is linked by State3 of the voice user and State3 of the SysUser
SET_MAP.TOPCALL.TCVMail.STATE.S.State2.Name;TCVMAIL Greeting2,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State2.Flags;1,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State2.INT_PROMPT;.\remote2.wav;att
SET_MAP.TOPCALL.TCVMail.STATE.S.State2.EXT_PROMPT;.\remote3.wav;att

//Is linked by State8 of the voice user
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Name;TCVMAIL Schedule1,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Flags;1073741824,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object1.Days Of Week;124,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object1.Start_Time;8:00,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object1.End_Time;17:00,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object1.Voice_ProfileID;State1,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object2.Days_Of_Week;127,int
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object2.Start_Time;12:00,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object2.End_Time;13:00,str
SET_MAP.TOPCALL.TCVMail.STATE.S.State3.Schedules.Schedule_Object2.Voice_ProfileID;State2,str

```

9.3.3.20 Importing Voice Prompts

Some users already have a pre-recorded greeting which they would like to use with VA. With the TC/Usertool, the administrator or the user can import this prompt into the profile of the user.

9.3.3.20.1 Prerequisites and Additional Installation

See [\[18\]](#)

9.3.3.20.2 Configuration

See [\[18\]](#)

9.3.3.21 Alternative Telephone Number and Representative

By using [The Attendant](#), the user can specify a alternative number and a representative. When a caller is prompted to leave a message he will hear: "To leave a message press 1. To call the mobile phone, press 2. To call the representative press 3 ...".

9.3.3.21.1 Prerequisites and Additional Installation

The services "VOICE_K2" and "VOICE_K3" are necessary. The services are of the type "FREE".

9.3.3.21.2 Configuration

The alternative number can be configured for each user separately as service "VOICE_K2". If VA finds such a service, the caller will have the possibility to call the number of the service.

The representative's number can be configured for each user separately as service "VOICE_K3". If VA finds such a service, the caller will have the possibility to call the number of the service.

You can customize the name of the service used for the alternative number with the registry key "ServiceAttendant2". The registry key for the service used for the representative is "ServiceAttendant3".

Note: Since version 3.08.01 of VA, the user can enable/disable this feature via TC/Web (version 4.08.04)

The administrator can disable the attendant individually for each user. Therefore, he has to add the following values to the "flags" field of the current active state of the user's profile.

- The value 16 disables the alternative number.
- The value 32 disables the representative.

E.g.: The user's current active state is "STATE4" and the flags are already set to 2. The administrator wants to disable the representative. So he must add 32 to 2 and gets 34.

```
tcusertool login:user;pwd filter:ts_user_id;user cmd:change /
set:SET_MAP.TOPCALL.TCVMAIL.STATE4.FLAGS;34;int
```

9.3.3.22 Disable/Enable "This message contains no attachments"

After VA has played a message, it tells the user if the message contains any attachments.

When the messages did not contain any attachments it is possible to play "This message contains no attachment" or to skip this prompt.

9.3.3.22.1 Configuration

If the key "TCVMAIL\PlayEmptyAttachmentList" is "1", the prompt "This message contains no attachment" will be played. If set to "0", this prompt will be skipped.

9.3.3.23 TCSI Codepage Conversion

Although the standard TCOSS codepage does cover a lot of languages, some of them need different codepages.

9.3.3.23.1 Related Topics

[TCSI Codepage conversion \(VL\)](#)

9.3.3.23.2 Configuration

If the following keys are not already created during setup, create them.

```
"TCVMail\AnsiCodePage" [REG_DWORD]
"TCVMail\TCOSSCodepage" [REG_DWORD]
```

VA will allow every value from 0 to 9999, with 0 as default, but only valid TCOSS codepages must be entered.

E.g.: When sending messages with a Greek character set, both values have to be set to "1253". For a Japanese installation, set both values to "932"

For details about codepage conversion please refer to the TC/Link manual page 29.

9.3.3.24 Greeting Wizard

The Greeting Wizard [GWIZ] is invoked, the first time a user logs into his mailbox.

In the current implementation the GWIZ will simply ask the user for his name and his greeting and store them as the active VP. After the user recorded his greeting, the wizard is deactivated automatically and the user can proceed to listen to his mailbox.

9.3.3.24.1 Prerequisites and Additional Installation

The TC/UserTool is needed and will be installed with the KCS (7.59.00 or higher) if the menu-item TC/VoiceAccess was selected.

9.3.3.24.2 Configuration

The GWIZ can be enabled or disabled system wide with the registry key "TCVMAIL\EnableConfWizard" (DWORD). To enable the GWIZ set the key to "1". To disable the GWIZ set the key to "0".

Per default, the GWIZ is disabled and must be activated separately for each user.

Therefore, change the item "SET_MAP.TOPCALL.TCVMail.Settings.ConfigWizard" in the user's profile to "2" (integer).

If this value is set, the GWIZ will query the user, for his name and greeting.

When the user logs into his mailbox a second time, he will not be asked again. To reactivate the GWIZ again, reset the value to "2".

E.g.:

```
tcusertool login:Server;UserId cmd:change filter:ts_user_id;UserID
set:SET_MAP.TOPCALL.TCVMail.Settings.ConfigWizard;2;int
```

9.3.3.25 Lock Filter

This feature allows the administrator to deny a user to change his filter settings.

9.3.3.25.1 Configuration

Currently this feature can only be configured with the TC/Usertool.

First set the location and the name where the lock can be found in the user's profile.

To do this change the registry key "TCVMAIL\AllowFilterChange" [REG_SZ] to a valid path in the user profile. The default is

"UN_CONTENT.SET_CLIENT_SETUP.SET_MAP.TOPCALL.TCVMail.SETTINGS.AllowFilterChange"

If this path does not exist, VA will allow the user to change the filter.

If this path exists and is set to <> "1", VA will deny the user to change the filter.

9.3.3.26 Call Duration Logging

VA creates log entries in TCOSS to allow a detailed reporting and billing of incoming call.

Note that a call may have more than one log entry. This is done because a call may get transferred to a different number and back again. (E.g.: When using [CallSender](#))

A log entry is of type "VOICEACCESS_LOG" and has the following fields:

- SESSION_ID = A unique ID that is the same for all log entries during a call. The value is created out of the current date, time and the handle of the call.
(E.g.: 2005-02-24 13:59:47:-2147418091")
- LOG_TYPE = Indicates the type of log entry.
0 = the caller called to leave a message
1 = the caller called to listen to his voicemails
2 = the caller did a "calltransfer" to the number specified in "TRANSFER_NR"
- RECIPIENT_NR = The number the caller called. When the LOG_TYPE was "1", this number is the access number of the voice server.
- ORIGINATOR_NR = The number of caller
- TRANSFER_NR = 'When number of the "calltransfer" destination. (E.g.: The number of the operator, representative or the originator of a voicemail)
- COSTCENTER = The cost center of the called number. It is not the cost center of the calling person!
- START_TIME = The date and time when the call was received.
- DURATION = The duration of the logged part of the call in seconds. To get the complete duration of the call, summarize the duration of all calls with the same SESSION_ID.

9.3.3.26.1 Configuration

Per default the logging is enabled but can be disabled via the registry key "TCVMAIL\doLogging". Set it to "0" to disable logging.

9.3.3.27 Enhanced Distinction of External and Internal Callers

Background information:

A user can record two different greetings, one for internal callers and one for external callers. Every caller whose caller id is also a voice address will be treated as internal caller. All the other callers will be treated as external callers. This is sufficient if all the employees of a company have an address book record with a voice address on TCOSS. But there are cases where some or most employees do not have their own mailbox but should also hear the internal greeting. For these cases this feature was implemented.

When the caller id of a caller is shorter or equals a configurable length, the internal greeting will be played.

9.3.3.27.1 Configuration

The registry key “TCVMAIL\InternalNrMaxLength” [DWORD] specifies the maximum length of an internal telephone number. If a telephone number is longer than the value of the key, it is assumed that the call is external.

Note: This is the same registry key as in the feature “[CallSender](#)“ and has the same purpose there.

9.3.3.28 Customized Subject

Sometimes the default subject of a voice mail is not sufficient. VA provides a way to customize the subject.

9.3.3.28.1 Configuration

VA creates the subject out of several different parts.

First there is the “Voice mail from” part. This text can be changed with the key “Subject”

The next part is the caller id. Depending on the available information VA inserts the full name of the caller or only the caller id. If VA does not know the caller id, it will insert “UNKNOWN”. This “UNKNOWN” text can also be changed with the registry key “TCVMAIL\OriginatorAddress”.

Finally VA appends the duration of the message. The duration is put between brackets and the time-unit is appended. E.g.: [15sec] VA allows customization of the brackets (“TCVMAIL\SubjectBeginBracket” and “TCVMAIL\SubjectEndBracket”) and the time-unit string (“TCVMAIL\SubjectSeconds”). Note that changing the time-unit string will not change the value of the duration. VA will always calculate in seconds.

9.3.3.29 Account Locking

Account locking is a technique for intrusion detection and prevention. It is tied to the TCOSS account locking feature and uses the same techniques. This means that if a user enters a wrong pin too often not only VA will lock his account but also TCOSS. This also works vice versa. The administrator can unlock the account with TCfw.

Note: Users using the full-auto-login can still access their mailbox even if it is locked.

9.3.3.29.1 Configuration

VA distinguishes between the check if a mailbox is locked and that it should lock a mailbox.

To enable the check if a mailbox is locked, set “TCVMAIL\CheckMailBoxLock” to “1”. “0” disables the feature.

To tell VA to increment the lock counter in the TCOSS user profile after an unsuccessful login attempt, set the key “TCVMAIL\LockMailBox” to “1”. “0” disables the feature.

Because VA uses the mechanism of TCOSS for account locking, the rest of the configuration has to be done in TCOSS.

9.3.3.30 Password Expiry

The password expiry enforces a strict password policy in VA. If enabled each user has to change the password after a specified period of time. The user cannot access his mailbox until he changed the password.

9.3.3.30.1 Related Topics

[Deny Continuous Number For The PIN](#)

9.3.3.30.2 Configuration

The period after which the password expired is defined with the key "TCVMAIL\PinExpiry". The value of this key is the period of days the password is valid. "0" disables the feature.

If activated, VA will remember when a password was changed the last time and prompts the user for a new password after the configured period expires.

Three days before the password expires VA reminds the user to change the password.

A typical scenario can look like this.

- User calls VA.
- The user enters his mailbox number and password.
- VA prompts the user that his password will expire in two days.
- The user can proceed and listen to the messages in his mailbox.
- After two days the user calls VA again.
- He reenters his mailbox number and password.
- This time VA prompts the user that the password has expired and that he has to enter a new one.
- If the user refuses to do so, VA will hang up. And the same procedure will be started the next time the user calls.
- If the user enters a new valid password, the password will be saved and he can now proceed to listen to the messages in his mailbox.

9.3.3.31 Password Length

The length of the password to the mailbox is configurable in the registry keys:

- TCVMAIL\PwdMin (default value: "2", valid values "1" – "4")
- TCVMAIL\PwdMax (default value: "4", valid values "2" – "12")

If the minimum length is greater than the maximum length, the password length must exactly match the minimum length.

9.3.3.32 Tracing

VA can do quite heavy tracing and therefore it should be only activated for diagnostic reasons.

Every line traced by the VA starts with "VML:".

9.3.3.32.1 Configuration

Although there are more keys starting with "tracelevel", only the following three are of interest:

"TCVMail\Tracelevel_VM" [DWORD] controls the amount of trace of VA. For debugging, set it to "ff".

"TCVMail\Tracelevel_TFC" [DWORD] enables message dumps when set to "ff". The dumps are stored in "C:\tcoss\trace".

"TCVMail\Tracelevel_STM" [DWORD] controls the amount of trace of the state machine. For debugging, set it to "ff".

"TCVMail\Tracelevel_Greet" [DWORD] controls the amount of trace of the greeting sub system. For debugging, set it to "ff".

9.3.3.33 Integration with Kofax Communication Server Archive

If a mailbox owner logs on to his voice mailbox via telephone and sends messages to SPI users (via forward, reply, reply to all, compose new message), these messages are normally not stored in the Kofax Communication Server Archive (as they are directly sent to a mail server).

9.3.3.33.1 Configuration

Archive integration can be enabled via the registry key "\TCVMAIL\BCCRecipient" [REG_SZ].

The Kofax Communication Server user specified in this registry key is added to the message as a BCC recipient.

If VA is in use with the TC/SPI-MX, the BCC recipient must be the same user as configured in the TC/LINK-MX7 registry (*TOPCALLArchiveName*). Thus, the message is forwarded to TC/LINK-MX7 and is archived via the standard TCLINK algorithm.

9.3.3.34 Deny Continuous Number for the PIN

This feature was implemented for security reasons. If activated, changing the PIN code on the telephone using a number with sequential digits is not allowed. Also a number without varying digits is not allowed.

VA will not allow any sequence of digits where the maximum increasing or decreasing is not bigger than 1.

Numbers like the following cannot be used:

- 1111 (no increasing or decreasing at all.)
- 1234 (increases by 1)
- 4321 (decreases by 1)
- 1112 (the maximum interval is still 1)
- 1212 (increases and decreases by 1)

However, following numbers are allowed:

- 1311 (the maximum interval is 2)
- 1231 (again, the maximum interval is 2)

In addition the mailbox number as a PIN code is also forbidden.

9.3.3.34.1 Related Topics

[Password Expiry](#)

9.3.3.34.2 Configuration

To activate the feature set the registry key “TCVMAIL\EnhancedPwdCheck” [DWORD] to “1”.

9.3.3.35 Logging of Invalid Login Attempts

This feature is an enhancement of an already existing functionality. In the case of an invalid login attempt, it does not only create an event log entry but also creates a log entry in TCOSS.

- The log entry in TCOSS consists of the following fields:
- The string “VOICEACCESS_FAILED_LOGIN”
- The caller id
- And the login time

9.3.3.35.1 Related Topics

[Call Duration Logging](#)

9.3.3.35.2 Configuration

Like [Call Duration Logging](#) the feature can be enabled by setting the registry key “TCVMAIL\doLogging” [DWORD] to “1”.

9.3.4 TC/PreProcess

An [introduction of TC/PreProcess](#) can be found in *Overview and Concepts chapter*. This section explains how to configure the features of TC/PreProcess which are

- MetaMail resolver unit (processes MetaMail URLs and removes inserted help texts)
- Watch Thread (repairs connection to TCOSS and archive, reloads configuration)
- Replacer unit (doing simple text mappings)
- Converter unit (performing text conversions with wildcards)

TC/PreProcess is installed as a [part of the VA setup](#) which is described in the *Installation - Setup chapter*. It must be installed during the setup. No additional installation is needed.

For information about *MetaMail* setup, installation and configuration read [\[12\]](#).

9.3.4.1.1 Configuration

9.3.4.1.1.1 Activating TC/PreProcess

To activate the TC/PreProcess, enable the email reader [VA basic setup screen](#) and then check *Enable the preprocessor* in the following setup screen. You can also set the registry key TCVMail\doPreProcessing to “1”.

9.3.4.1.1.2 General Parameters

TCPrep\DoReplace (REG_DW)

Completely deactivates the replacer unit of TC/PreProcess if set to “0”. This can be used for enhancing the performance, by manually disabling all replacer actions. It can sometimes also help for troubleshooting.

TCPrep\DoConvert (REG_DW)

Same as DoReplace, but for the Converter unit.

TCPrep\DoMetaMail (REG_DW)

Same as DoReplace, but for the MetaMail resolver unit.

TCPrep\StopAfterCharacters (REG_DW)

This is a limit for the maximum message length in characters which is processed. If this limit is reached during processing, TC/PreProcess will stop and return to the controlling instance (mostly TC/VoiceAccess). This is a performance protection to prevent TC/PreProcess from processing kilobytes of text. Therefore this value should be set to high limits (like the default of 10000). Note that this limit is currently only checked before each line that is processed. TC/PreProcess fetches one line (= text separated by newlines), evaluates the length of this line, adds it to the already processed characters count (from already processed lines) and checks if this is below the StopAfterCharacters value. If it is not, the line will not be processed anymore and TC/PreProcess returns to controller. If now for instance the first line of a mail has 60 characters and a (senseless) value of 10 is entered for StopAfterCharacters, nothing of the message, not even the first line, will be processed.

9.3.4.1.2 Related Topics

To understand the Replacer and Converter, it is important to know that TC/PreProcess works token-based. [Tokens cannot contain whitespace](#) and other characters; this is explained in the Restrictions chapter.

9.3.4.2 Tracing

9.3.4.2.1 Recommended Trace Level

The full trace level (0xffffffff) is not recommended, since it slows down the system very much and contains too much information for normal troubleshooting.

General problems can be found with 0x13 (error-, warning-, info- and data-trace).

There are unit-specific trace levels for the replacer, the converter and the MetaMail resolver. They are used to find problems which are specific to a certain functionality. If you encounter problems with MetaMail resolving, it is suggested to activate the MetaMail resolver trace.

If you are in doubt, activate 0xffffffff3 for troubleshooting.

9.3.4.2.2 Applying Trace Level Changes

The trace level is read from the registry every time the process function is called which happened every time a message is played via phone. In idle mode, it is read every [Watch Thread](#) interval which is 300 seconds.

9.3.4.3 MetaMail Resolver

If TC/PreProcess finds a MetaMail specific URL, it will find the message via the URL and insert it into the message body. The preprocessing continues at the beginning of the inserted message. If the message cannot be found in the message store it will be searched in the archive, if there is one.

Encrypted URLs

The MetaMail URL can contain encrypted parameters, to protect user and message details. An encrypted URL's parameters start with "EnId=" followed by the encrypted parameters. TC/PreProcess decrypts the string and processes the parameters. Parameters after the "EnId=...&" are ignored. To generate encrypted MetaMail URLs, TC/LP 2.09.03 or higher is required. Refer to TC/Link and TC/At2URL documentation for details.

Note: TC/PreProcess can only resolve MetaMail which means that it is only able to resolve a MetaMail URL for reading the message, but not to create a MetaMail URL (sending). MetaMail URLs are generated by the Link. To generate messages containing MetaMail URLs (MetaMail messages), it is necessary to have the **TC/LP (2.05.08)** installed and **TCA2URL** configured properly. For the use of MetaMail TC/LP is also required. See the MetaMail documentation for details.

9.3.4.3.1 Removing of PreURLText

If a MetaMail message (a message sent to e.g. a Lotus Notes mail system with a URL referring to the original message on the Kofax Communication Server) is generated by the link, the link exit TC/At2URL will insert a help text before the URL by default. This help text is configured in the registry values PreURLText and PreURLTextForTermination in the TCA2URL subkey of the link configuration in the registry (refer to Link and TC/At2URL documentation for details.). This makes MetaMail messages look like the following.

```
Use the following link to view the attachments of the message:  
http://demotc/LOGIN.ASP?UserId=CHKA&MsgCifId=16512722&MsgId=00FBF6D115566C71&MsgTime  
Intend=040506:082313
```

Every line in the processed message which matches exactly to the strings specified in these values gets removed. **If the values are empty ("") the feature will be disabled and nothing will be removed.** These values must be set to the same setting as values with the same name in the TC/At2URL-configuration to delete such lines inserted by TC/At2URL.

TC/PreProcess creates the values. It suggests "**Use the following link to view the attachments of the message:**" for PreURLText and "**It is necessary to click on the following link to terminate the message:**" for PreURLTextForTermination. These are the same strings which are used by TC/At2URL by default.

If TC/PreProcess finds out that the values do not exist at starts up (at TCECP startup), it will create the values with empty strings (remove nothing).

Attention: This facility deletes any line in the message if it matches exactly to the string specified in the values PreURLText and PreURLTextForTermination. **This may also erase lines which are not intended to be erased.**

9.3.4.3.2 Configuration

TCPrep\HttpAddress (REG_SZ)

Contains the name or IP address of the server occurring in the MetaMail URLs (the web server). TC/PreProcess uses this string to detect if a URL is a MetaMail URL. Upper-/lowercase is ignored. Make sure to include the leading "http://" and the trailing "/login.asp" here. This value must contain the same data as the value HttpAddress in the configuration of the link which inserts the MetaMail URLs in the mails.

TCPrep\MMServerName (REG_SZ)

Contains the hostname or IP address of the Kofax Communication Server where TC/PreProcess should try to get the messages, referred by MetaMail URLs, from.

TCPrep\MMServerPath (REG_SZ)

Contains the TCTI server path, including linktype and hostname or IP address, of the Kofax Communication Server where TC/PreProcess should try to get the messages, referred by MetaMail URLs, from. If the TCOSS server has the name mytcoss this value has to be set to "tcp/ip,mytcoss".

TCPrep\MMServerUserId (REG_SZ)

The user id, which is used to log on to the TCOSS server, is specified in MMServerName and MMServerPath. This user must have sufficient rights to view all users inboxes.

TCPrep\MMServerUserPwd (REG_SZ)

Password of the user specified in MMServerUserId. If this field is once read by TC/PreProcess it gets encrypted and is written back to the registry.

TCPrep\PreURLText (REG_SZ)

Text which can be inserted by TC/At2URL before MetaMail URLs. TC/PreProcess removes lines, which match exactly to the string specified in this value. Should be set to the registry value with the same name in the configuration of TC/At2URL, which generates the MetaMail messages.

TCPrep\PreURLTextForTermination (REG_SZ)

Same as meaning as PreURLText. TC/At2URL can insert PreURLText or PreURLTextForTermination before URLs. TC/PreProcess can remove both.

Since MetaMail search is also done in the archive of a Kofax Communication Server and the Kofax Communication Server Archive can also run on another machine, the last 4 values described here do also exist for the Kofax Communication Server Archive. These are: **TCPrep\MMArchServerName**, **TCPrep\MMArchServerPath**, **TCPrep\MMArchServerUserId**, **TCPrep\MMArchServerUserPwd**. Archive servers are usually specified with endpoint constants like "tcp/ip,mytcoss:archive" in MMArchServerPath for instance. If no archive server configuration is given no MetaMail search in archive will be done. **If the one of the MetaMail servers cannot be reached, a long delay can occur on initialization of TC/PreProcess**. If no MetaMail server can be reached, the feature will be disabled.

All values for MetaMail configuration are empty after being created by TC/PreProcess.

A MetaMail configuration where message server, archive server and the web server run on the same machine which has the hostname "tom", could look like this:

Name	Type	Data
ab(Default)	REG_SZ	(value not set)
abHttpAddress	REG_SZ	http://tom/login.asp
abMMArchServerName	REG_SZ	tom:archive
abMMArchServerPath	REG_SZ	tcp/ip,tom:archive
abMMArchServerUserId	REG_SZ	tcprepuser
abMMArchServerUserPwd	REG_SZ	28383B3F2C0407
abMMServerName	REG_SZ	tom
abMMServerPath	REG_SZ	tcp/ip,tom
abMMServerUserId	REG_SZ	tcprepuser
abMMServerUserPwd	REG_SZ	28383B2D3B0107
abPreURLText	REG_SZ	Use the following link to view the attachments of the message:
abPreURLTextForTermination	REG_SZ	It is necessary to click on the following link to terminate the message:
bbStopAfterCharacters	REG_DWORD	0x00002710 (10000)
bbTracelevel	REG_DWORD	0xffffffff (4294967280)

To access the Kofax Communication Server, TC/PreProcess uses the TFC (TC Foundation Classes). **A TFC license must be installed on the Kofax Communication Server and Kofax Communication Server Archive.**

The MetaMail resolver and PreURLText removing features can be completely switched off by creating a REG_DW registry value named **DoMetaMail** in the TCPREP key and setting it to 0.

9.3.4.4 Watch Thread

When TC/PreProcess is initialized by VA which happens when TCECP starts, it starts a so-called Watch Thread.

The Watch Thread will read all registry values and checks if the connections to the TCOSS and archive servers are still available. This avoids that the server, due to reaching the session idle timeout, closes the sessions automatically. The Watch Thread does this every 300 seconds.

If a session is lost, the watch thread will try to reconnect. If the lost session is detected, a TCTI server timeout will occur which makes the Watch Thread hang for some seconds. A call of the process function (playing a message) will have to wait until the Watch Thread came over the TCTI server timeout. It can happen that the Watch Thread tries to check the connection while a MetaMail URL is resolved. In this case, the Watch Thread will skip the check of the TCOSS or archive server connection.

The Watch Thread has its own trace level where the activity can be monitored. See description of the registry value TCPREP\TraceLevel for details.

9.3.4.5 Replacer

TC/PreProcess has a replacer unit which looks up every word in the file with the mappings and replaces it if a match is found. If a match is found, further mappings will not be compared with the actual word.

9.3.4.5.1 Configuration

A mapping file realizes some of TC/PreProcess' features. It is a text file in c:\topcall\shared named **tcprep.mpf** containing the mappings. Every line contains one mapping and consists of the following syntax.

```
"searchstring" "replacestring"
```

TC/PreProcess fetches a **token (a word separated by whitespace, one of: “.,?!” or newline)** from the message and compares it to the searchstring of every line in tcprep.mpf until a match is found. If a match is found, the token will be replaced by replacestring. **The comparison is case sensitive and stops after the first match.**

Mappings could look like “TC”, “Topcall” or “replace_this_one”, “with_this_one”. With a leading ‘\’ also a quote character (“) can be used in the string specification of a mapping.

Mappings can also be used for removing specific strings like this:

```
“ignore” “”
```

Comment lines start with a doublecross (#) symbol.

The replacer feature can be completely switched off by creating a REG_DW registry value named **DoReplace** in the TCPREP key and setting it to 0.

If you have defined own mappings you have to set the file tcprep.mpf read-only to avoid losing your changes on next setup!

9.3.4.5.2 Limits

A maximum of 200 mappings is allowed. A search- and replace string in mappings may not be longer than 300 characters.

9.3.4.5.3 Example for tcprep.mpf

```
“FAQ:”      “frequently asked questions”
“FWD:”      “forwarded”
“FW:”        “forwarded”
“ASAP”       “as soon as possible”
“CU”         “see you”
```

9.3.4.5.4 Resolving Common Abbreviations

Abbreviations commonly used or known can be replaced with proper mappings like in the example of tcprep.mpf.

9.3.4.5.5 Replacing Typographic Smileys

Typographic smiley's as “:)” or “:-)” can be replaced by appropriate words like “ha ha” or “boo-hoo” with proper mappings.

9.3.4.6 Converter

The converter unit of TC/PreProcess applies complete rules to each token and if the rule matches, a proper conversion will be done. Note that replacement operations are much faster than rule processing for conversions. This should be considered when creating rules and mappings. **Before a token gets to the converter unit, it passes the replacer unit. Once replaced, the token will not get to the converter anymore.** The comparison with rules in the converter unit is, like in the replacer unit, **case sensitive**.

9.3.4.6.1 Configuration

The conversion is defined by specified rules. **The syntax of the rules is similar to the syntax of the mappings, with the only difference, that wildcards may be used.** The rules are kept in a file called **tcprep.rul**. The difference is that rules allow wildcards. Wildcard symbols are escaped with a \ before its symbol. This ensures that all printable characters can be used in a text part of a rule (like "*").

Wildcards

The following wildcards may be used.

Symbol	Meaning
*	0 or more unknown characters
?	0 or one unknown character
.	One unknown character

Table 13 – TC/PreProcessor Wildcards

For example a rule like:

```
"*CFO**" "* chief financial officer *"
```

would convert a text like "(CFO)" into "(chief financial officer)".

It is also possible to remove parts of the string by specifying less text parts or wildcards on the right side than on the left:

```
"*letmefree*" "free"
```

This rule would convert strings like "212letmefreedfh" or "letmefreenow" into "free".

Wildcard text parts can only be deleted at the begin or at the end of a rule.

```
"begin*end*" "beginend*"
```

This rule would turn "begin111end222" into "beginend111" because the first "begin" would be replaced with "beginend" (replace text with text) then, the first wildcard part of the input text ("111") would be inserted.

The end of the right part of the rule would be reached, so the rest of the input text would be erased.

The converter feature can be completely switched off by creating a REG_DW registry value named **DoConvert** in the TCPREP key and setting it to 0.

If you have defined own rules you have to set the file tcprep.rul read-only to avoid losing your changes on the next setup!

9.3.4.6.2 Limits

A maximum of 200 rules is allowed. Left and right strings in rules may not be longer than 300 characters each.

9.3.4.6.3 Example for tcprep.rul

```
" :* )"      "smiley face with or without nose"
" *- )"     "smiley face with unknown eyes"

# The next rule removes the braces from a text,
# where the text in braces is a token as explained in the chapter replacements
"( *)"      " **"

# The next rule would convert a text like "(CFO)" as well as "CFO",
# which would not be possible with replacements
```

```
“*CFO*” “chief financial officer”
```

9.3.4.6.4 Removing Notes Domain Names from mails

With proper rules TC/PreProcess removes all notes domain names from the following email.

```
Might be interesting for you

regards,
Nastasja

----- Forwarded by Nastasja Plohn/TC/TOPCALL on 22.10.2003 13:44 -----
klaus.winter@ittraining.at
21.10.2003 11:23
To: Nastasja Plohn@TOPCALL,
cc: John Miller/TC/TOPCALL@TOPCALL, Ben Croft/UK/TC/TOPCALL@TOPCALL, Juan
DePalma/ES/TC/TOPCALL@TOPCALL
Subject: NewsFlash! Introduction to Voice over IP - last Chance: 13th Nov.!

20th October 2003

In this issue...
TCP/IP:
Introduction to Voice over IP

Contact:
Your Trainer: Peter Sommer
```

The result looks like this. The mail is better understandable for the user, because the (mostly identical) Lotus Notes domain names are not read by the TTS.

```
Might be interesting for you

regards,
Nastasja

----- Forwarded by Nastasja Plohn on 22.10.2003 13:44 -----
klaus.winter@ittraining.at
21.10.2003 11:23
To: Nastasja Plohn
cc: John Miller, Ben Croft, Juan DePalma
Subject: NewsFlash! Introduction to Voice over IP - last Chance: 13th Nov.!

20th October 2003

In this issue...
TCP/IP:
Introduction to Voice over IP

Contact:
Your Trainer: Peter Sommer
```

Rules like these are necessary for such a conversion.

```
# Remove notes domains

“*@TOPCALL*” “***”
“*/TC/TOPCALL*” “***”
“*/TC/TOPCALL@TOPCALL*” “***”
“*/UK/TC/TOPCALL@TOPCALL*” “***”
“*/ES/TC/TOPCALL@TOPCALL*” “***”
```

9.3.5 TC/VoiceXML

9.3.5.1 Built-In Grammars

Additionally to the standard VoiceXML built-in grammars, TC/VoiceXML allows the definition of additional grammars which are treated like built-in grammars.

9.3.5.1.1 Configuration

The additional built-in grammars are stored in an XML file named “builtin-dtmf.xml” in “C:\tcoss\system”.

Its location and name can be configured in the config-file with the line

```
client.userdefined.grammar    VXIString file://C:\\TCOSS\\SYSTEM\\builtin-dtmf.xml
```

This line is not written by default.

For details about how a built-in grammar file can be edited refer to [\[1\]](#).

9.3.5.2 Tracing

The tracing in TC/VoiceXML can be controlled in very detailed way. TC/VoiceXML allows to set trace levels separately for each part of the implementation.

9.3.5.2.1 Configuration

In the config file c:\tcoss\system\tcvxiclient.cfg, every module has its own identification number:

client.cache.diagLogBase	VXIIInteger	2000
client.inet.diagLogBase	VXIIInteger	3000
client.jsi.diagLogBase	VXIIInteger	4000
client.prompt.diagLogBase	VXIIInteger	5000
client.rec.diagLogBase	VXIIInteger	6000
client.tel.diagLogBase	VXIIInteger	7000
client.vxi.diagLogBase	VXIIInteger	8000
client.object.diagLogBase	VXIIInteger	9000
client.client.diagLogBase	VXIIInteger	10000

This numbers must not be changed!

With the knowledge of this numbers (e.g.: Prompter is 5000), it is possible to enable or disable certain tracelevels.

```
#####
# Diagnostic tags for the prompt interface
# 0 to disable, 1 to enable
# 5000 API/general
# 5001 Prompting
# 5002 Prefetching
# 5003 Properties
#####
client.log.diagTag.5000      VXIIInteger 0
client.log.diagTag.5001      VXIIInteger 0
client.log.diagTag.5002      VXIIInteger 0
client.log.diagTag.5003      VXIIInteger 0
```

Changing the value of the key “client.log.diagTag.5003” to “1” would enable the “properties” trace of the prompter. Now every time the prompter has to play a prompt, all properties passed are traced. To disable the trace set it back to “0”.

You will find a description of the possible trace settings for every module in the config file.

9.3.5.3 Defaults

Because TC/VoiceXML has a lot of properties which does not need to be set for each VoiceXML application again, a default file exists. This file is always loaded before the actual VoiceXML application is loaded and should therefore only contain defaults which are globally used and useful.

The configuration only gives a short overview. For more details refer to [\[1\]](#)

9.3.5.3.1 Configuration

This file can be found under “C:\TCOSS\SYSTEM\Config” and is named “defaults.xml”.

It is not only a list of properties but also a list of VoiceXML elements which are used at the root scope of the VoiceXML application.

E.g.:

```
<noinput count='2'>
    <prompt xml:lang="en">Sorry, I didn't hear you.</prompt>
    <reprompt/>
</noinput>
```

These lines enable the second re-prompts in the “noinput” condition, but more interesting are the possible properties and their meaning:

```
<property name='fetchaudiodelay' value='2s' />
```

Becomes active if TC/VoiceXML has to fetch a “distant” VoiceXML file. In such a case an audio file can be played (E.g.: “Please wait” followed by some melody). The fetchaudiodelay tells to start playback after two seconds if the requested file is still not ready.

```
<property name='fetchaudiominimum' value='5s' />
```

The next property related to fetching a VoiceXML file which is active after the fetchaudiodelay. In this case, the playback of the audio file was started. Fetchaudiominimum specifies during which period the playback must not be interrupted. This is done to prevent situations where only small parts of the audio file are played (E.g.: “Ple..” followed by the next prompt).

```
<property name='fetchtimeout' value='7s' />
```

The general timeout how long TC/VoiceXML should wait for a VoiceXML file. If this time exceeds, the execution will be interrupted by an error. This error can be handled by using a “<catch>“ element.

```
<property name='bargein' value='true' />
```

Sets “bargin” for prompts to the default value “true”.

```
<property name='bargeintype' value='speech' />
```

Allows DTMF and speech “bargin”.

```
<property name='timeout' value='5s' />
```

This timeout is used if TC/VoiceXML waits for an input. If, after this timeout, the user did not provide an input, the “noinput” event will be thrown.

```
<property name='interdigittimeout' value='2s' />
```

The interdigittimeout is used during the DTMF input. In this case the user has two seconds time between each DTMF before a “NOMATCH” event will be thrown.

```
<property name='termtimeout' value='0s' />
```

This timeout is of special interest when complex grammars are used. It is active when the user provided an input that already has a matching grammar. If the termtimeout is zero seconds, TC/VoiceXML will throw a "match" immediately. If the termtimeout is greater than zero the user has the possibility to add additional input. Consider following scenario.

The possible inputs is: "123" or "1234". In the case where termtimeout is zero, the user is never able to enter "1234" because as soon as he enters "123" "match" is thrown.

```
<property name='termchar' value='#' />
```

The termchar is the DTMF which terminates the input of the user. The termchar is not added to input but triggers the final grammar matching. This is useful when the user can enter a variable length input. The user then simply terminates his input by pressing the termchar.

Considering the above scenario this time with a termtimeout of five seconds.

When the user enters "123" either he has to wait five seconds before his input is processed or he presses "123#" and the input is immediately processed.

There are some additional proprietary properties which can be set:

```
<property name='com.topcall.loopdetection' value='SBOX' />
```

Is the type of the loopdetection to perform. Possible values are "SBOX" and "PBOX". See [Call Transfer And Loop Detection](#) for details.

```
<property name='com.topcall.ctmode' value='INT' />
```

Refers to the call transfer mode. Possible values are "INT"(bridged) and "EXT"(blind). See [Call Transfer And Loop Detection](#) for details.

9.3.5.4 Answering Machine Detection

Because VoiceXML is also used with the VL2 the question comes up: "What to do if the VL2 reaches an answering machine?". The answer was to give the VoiceXML designer the choice. VoiceXML offers three possible responses if an answering machine is detected.

- 1.) Disconnect immediately. The current playback, recognition or recoding is stopped, and the connection is dropped. Normally, this happens fast enough so that no message is left. Only in some cases the mailbox software does not have a feature similar to [Minimum Recording Length](#). In this case a < 1 sec. message is left.
- 2.) Do nothing. If an answering machine is detected nothing is interrupted. Playback, recognition and recoding will continue until something else happens.
- 3.) Interrupt the current activity. The playback, recoding or recognition is interrupted immediately but the connection is not dropped. This will not leave any half-spoken sentences or long delays at the beginning of the voicemail and allows the VoiceXML designer to play a different prompt as soon as possible.

If an answering machine was detected during recognition or recoding a "no match" event is thrown. If the answering machine was detected during playback no special event occurs.

There is no "answering machine detected" event, but the VoiceXML designer must check the return value "detected" of the object

```
com.topcall.MailboxDetected
```

This method returns a variable called “detected” which has the value “true” in case of a successful detection of an answering machine. In all other cases the value of “detected” is “false”.

Note: Answering machine detection only works with LS1. If FoIPv3 is used, the value of “detected” is always “false”.

Attention: Due to the way the VoiceXML interpreter loads and processes the VoiceXML file, a simple check to “com.topcall.MailboxDetected” will always yield “false”. The reason is that the interpreter calls this method as soon as possible, and this is as soon as the file is loaded. With for e.g. a <subdialog> it is possible to prevent this early execution of the method.

A <subdialog> which asks for mailbox detection may look like this:

```
<property name="fetchaudiodelay" value="0.1s"/>
<subdialog name="result"
src="file:///c:/tcoss/system/mailboxdetection.vxml#mailboxdetection" fetchhint="safe"
fetchaudio="file:///C:/tcoss/system/silence.wav">

    <filled>
        <log>
            Mailbox detection returned : <value expr="result.result" />
            <log>
                <!-- Do something -->
            </log>
        </filled>
    </subdialog>
```

It is necessary that the called dialog is within a different file; otherwise the interpreter will call the “com.topcall.MailboxDetected” method too early.

Note: Please be sure to set the “fetchaudiodelay” to something small like “0.1s”. This ensures that the dialog returns as fast as possible and that the user will not hear any delays.

The called VoiceXML file may look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="2.0" xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd" xml:lang="en">
    <form id="mailboxdetection">
        <object name="retval" classid="com.topcall.MailboxDetected" />
        <block>
            <log>
                Mailboxdetection returned : <value expr="retval.detected" />
            </log>
            <var name="result" expr="retval.detected"/>
            <return namelist="result"/>
        </block>
    </form>
</?xml>
```

This code is taken from the file “mailboxdetection.vxml” which should be located in “c:\tcoss\system”.

Because of the two different situations where a mailbox can be detected, different approaches to check for it must be used. The easiest way is after a prompt.

```
<form id="myForm">
    <block name="Greeting1">
        <prompt>
```

```

        Hello. You are connected to an automatic voice message from
        Kofax Communication Server.
    </prompt>
</block>
<property name="fetchaudiodelay" value="0.1s"/>
<subdialog name="result"
    src="file:///c:/tcoss/system/mailboxdetection.vxml#mailboxdetection"
    fetchhint="safe" fetchaudio="file:///C:/tcoss/system/silence.wav">
<filled>
    <log>
        Mailbox detection returned: <value expr="result.result" />
    </log>
    <!-- Do something -->
</filled>
</subdialog>
</form>
```

This `<form>` would check for mailbox detection after the prompt “*Hello. You are connected to an automatic voice message from Kofax Communication Server.*” had been played.

When checking for mailbox detection within a `<field>` or `<menu>` does must be done within a `<nomatch>` element.

```

<menu id="menuHear" dtmf="true" accept="exact">

    <prompt> Please press something </prompt>
    <!-- Do something -->
    <nomatch>
        <goto next="#checkMailbox1"/>
    </nomatch>
</menu>

<form id="checkMailbox1">
    <property name="fetchaudiodelay" value="0.1s"/>
    <subdialog name="result"
        src="file:///c:/tcoss/system/mailboxdetection.vxml#mailboxdetection"
        fetchhint="safe" fetchaudio="file:///C:/tcoss/system/silence.wav">

        <filled>
            <log>
                Mailbox detection returned :
                <value expr="result.result"/>
            </log>
            <!-- Do something -->
        </filled>
    </subdialog>
</form>
```

Because `<nomatch>` does not allow an element `<subdialog>` as child-element it is necessary to perform a `<goto>` in advance. This only raises the problem that the re-prompting mechanism of the `<nomatch>` element is bypassed and has to be implemented on its own.

9.3.5.4.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The telephone system must be LS1

9.3.5.4.2 Related Topics

[VL2 - Answering Machine Detection](#)

9.3.5.4.3 Configuration

In the config file c:\tcoss\system\tcvxiclient.cfg, the object plug-in "tcvximailboxdet.dll" must be loaded:

```
client.object.plugin.2 VXIString C:/TCOSS/SYSTEM/TCVXIMailboxDet.dll
```

The "2" after "client.object.plugin" is used for enumeration only. So if "2" is already used, the next plug-in would be called "client.object.plugin.3". If "3" is already used the next plug-in would be called "client.object.plugin.4" and so on...

The behavior of the answering machine detection can be configured via the line

```
client.mailbox.detection.behavior VXIInteger 0
```

Possible values are:

0 ... Disconnect immediately (default)

1 ... Do nothing

2 ... Interrupt the current activity

9.3.5.5 Transfer the Voice Call to Fax

If connected through the LS1 lineserver, TC/VoiceXML is able to transfer any of its incoming voice calls back to the LS1 and let them be handled as a incoming fax call. This functionality can be useful for customers with limited numbering range for their fax extensions, they could handle their complete incoming fax traffic through a single public fax number using two-stage dialing like following:

1. Caller who wants to send a fax dials the customer's public fax number
2. This number is being handled as a voice number on the LS1, the call is answered by TC/VoiceXML
3. TC/VoiceXML prompts the caller to enter the fax number/extension
4. TC/VoiceXML transfers the call back to the LS1 which immediately starts with the fax transmission
5. The caller hears the fax prompts on the phone and also starts the fax transmission
6. The fax is received

In order to accomplish this functionality, call transfer's property com.topcall.ctmode must be set to the value 'FAX' and the parameter bridge must be set to "true":

```
<transfer name="transfer1" destexpr="'tel:' + extension" bridge="true">
<property name='com.topcall.ctmode ' value='INT' />
</transfer>
```

Please refer to the full VXML example for call transfer to fax:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxm version="2.0" xmlns="http://www.w3.org/2001/vxml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/vxml
      http://www.w3.org/TR/voicexml20/vxml.xsd">
<var name="extension"/>
    <form id="form1">
        <!-- x="34" -->
        <!-- y="194" -->
        <field name="field1" slot="number">
            <prompt count="1">
                This Script will forward your call to a fax extension.
            </prompt>
        </field>
    </form>
</vxm>
```

```

        Please enter a fax extension with a length of 3 or 4 digits:
    </prompt>
    <grammar src="builtin:dtmf/digits?minlength=3;maxlength=4"/>
    <filled>
        <assign name="extension" expr="field1" />
        <goto next="#form2" />
    </filled>
    <nomatch count="1">
        <prompt bargein="false">Invalid length of fax number.
        </prompt>
        <reprompt />
    </nomatch>
    </field>
</form>

<form id="form2">
    <!-- x="105" -->
    <!-- y="382" -->
    <block name="prompt1">
        <prompt>
            Transferring call to <value expr="extension"/>.
        </prompt>
    </block>
    <transfer name="transfer1" destexpr="'tel:' + extension" bridge="true"
    connecttimeout="10s" maxtime="0s"
    transferaudio="file://parkq.wav">
        <property name="com.topcall.ctmode" value="FAX"/>
    </transfer>
    <block name="prompt2">
        <prompt>
            Sorry, the transfer did not work, please try it again in
some
            minutes.
        </prompt>
    </block>
</form>
</vxml>
```

9.3.6 TC/VoiceLink2

9.3.6.1 Pollers

A poller is responsible for retrieving messages from a message store. Furthermore it informs the originator over the result of the call.

As already mentioned in [Concept and Architecture of the VL2](#) the VL2 can support different pollers which can be used simultaneously.

At the moment only the so called TSCI poller is supported.

9.3.6.1.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.1.2 Related Topics

[Concept and Architecture of the VL2](#)

9.3.6.1.3 Configuration

The pollers to use are configured with the registry key “TCVLink2\Pollers” [REG_MULTI_SZ]. The default is “tcv12polltcoss.dll=default”

Each line consists of the file name of the poller and its alternative name which is used as customer name when writing log entries. These two parts are separated by a “=”. E.g.:

```
tcv12polltcoss.dll=default
```

Note: It is not allowed to load the same poller twice, or use the same log entry name for more than one poller.

The TSCI poller is configured by default. The file name for the TSCI poller would be “tcv12polltcoss.dll”

9.3.6.2 TSCI Poller – TCOSS Configuration

The TSCI poller can be configured to use either the globally configured TCOSS, which is also used by TCECP for licensing, or have its own TCOSS configuration.

9.3.6.2.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The TSCI Poller must be configured as poller

9.3.6.2.2 Related Topics

[TCSI Codepage Conversion](#)

[Enhanced Message Termination](#)

[VL2 First Time Configuration - The TCOSS Server](#)

9.3.6.2.3 Configuration

Per default the TSCI poller reads TCOSS configuration from the sub-key “TOPCALL\”. The following registry keys are used:

“TOPCALL\Path” [REG_SZ] is the path to the TCOSS server (e.g.: “TCP/IP,MYServer”)

“TOPCALL\Server” [REG_SZ] is the name of the TCOSS server (e.g.: “MyServer”)

“TOPCALL\User” [REG_SZ] is the name of the user (per default “tcvmail”)

“TOPCALL\Internal” [REG_SZ] the password of the user.

It is possible to overwrite these settings with the registry keys

“TCVLink2\Path” [REG_SZ] is the path to the TCOSS server (e.g.: “TCP/IP,MYServer”)

“TCVLink2\Server” [REG_SZ] is the name of the TCOSS server (e.g.: “MyServer”)

“TCVLink2\User” [REG_SZ] is the name of the user (per default “tcvmail”)

“TCVLink2\Internal” [REG_SZ] the password of the user.

The TSCI poller will use this configuration as soon as at least the keys “TCVL2PollTcoss\Path” and “TCVL2PollTcoss\User” are not empty.

9.3.6.3 Telephone Number Prefixes

For convenience, it is possible that the TSCI poller adds a prefix to the destination telephone number automatically. The TSCI poller supports two different prefixes. The “dialout” prefix and the “internal” prefix. The “dialout” prefix is the prefix which should be used if the destination is located within the PSTN. In most cases this prefix is a simply “0” (zero). The “internal” prefix is necessary to reach recipients within the corporate telephone network.

9.3.6.3.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The TSCI Poller must be configured as poller

9.3.6.3.2 Configuration

The registry key “TCVL2PollTcoss\DiaOutPrefix” [REG_SZ] specifies the “dialout” prefix whereas the key “TCVL2PollTcoss\InternalPrefix” [REG_SZ] specifies the prefix to use for internal calls. Both keys are empty per default.

9.3.6.4 TCOSS Queue

Per default the TSCI poller will poll the TCOSS queue “VL” but this may be changed if needed.

9.3.6.4.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The TSCI Poller must be configured as poller

9.3.6.4.2 Related Topics

[TSCI Poller – TCOSS configuration](#)

9.3.6.4.3 Configuration

With the registry key “TCVL2PollTcoss\Queue” [REG_SZ] the queue which the TSCI poller should poll can be specified. The default value is “VL”.

9.3.6.5 Logging & Billing

The VL2 is capable of logging certain events which might also be used for billing purposes.

These events are:

- 1.) A new message was retrieved.
- 2.) The VL2 tries to establish the call.

- 3.) The call is established and the playback starts.
- 4.) The call is disconnected.
- 5.) The response is processed.

For each of these events, the VL2 creates a log entry with all required information.

This information includes:

- 1.) A unique message ID.
- 2.) If configured, the name of the customer.
- 3.) The time when an event occurred.
- 4.) The duration between the start of playback and disconnect in seconds rounded to the nearest integer value (6.4sec = 6sec. 6.5sec = 7sec).
- 5.) The called telephone number.
- 6.) The error code if an error occurred.
- 7.) The response from the caller.

At the moment there are two different implementations for a logger. The File-Logger writes the events to a file on the hard disk whereas the TSCI Logger uses the TCOSS logging facility. Per default the TSCI Logger is used.

Regardless of the used logger, the VL2 will also save the duration of a call directly to the message if the TSCI poller is used. This duration can be displayed with e.g. TCfW.

9.3.6.5.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.5.2 Related Topics

[VL2 First Time Configuration – The TCOSS Server](#)

[VL2 Pollers](#)

9.3.6.5.3 Configuration

The logger which should be used can be configured via the registry key “TCVLink2\Logger” [REG_SZ]. It holds to the filename of the desired logger. To disable the logger leave the value empty. This is also the default.

For the TSCI Logger enter the value “tcvl2logtcoss.dll”.

For the File-Logger enter the value “tcvl2logfile.dll”.

Depending on the chosen logger the following configuration possibilities are available:

1.) TSCI Logger:

With the registry keys “TCVL2LogTcoss\Internal” [REG_SZ], “TCVL2LogTcoss\Path” [REG_SZ], “TCVL2LogTcoss\Server” [REG_SZ] and “TCVL2LogTcoss\User” [REG_SZ] the configuration of the TCOSS server from the registry key “TOPCALL\” can be overwritten. For example, if it is desired to log to a different server then the server which is used for polling. In order for the TSCI Logger to use these values instead of the values from the “TOPCALL\” sub-key at least the keys “TCVL2LogTcoss\Path” and “TCVL2LogTcoss\User” must not be empty.

All above mentioned values are empty per default.

The key “TCVL2LogTcoss\Internal” defines the password to use.

The key “TCVL2LogTcoss\User” defines the user to use for login.

The key “TCVL2LogTcoss\Path” defines the path to the TCOSS server.

The key “TCVL2LogTcoss\Server” specifies the name of the TCOSS server.

2.) File-Logger

With the registry key “TCVL2LogFile\LogFile” [REG_SZ] the full path to the log file must be supplied. The default is “C:\TCOSS\SYSTEM\VL2.log”

Note: The customer name is configured per poller. To change it, see [VL2 Pollers](#)

9.3.6.6 Remote Voice Server

As described in [the concept of the VL2](#) it is possible that the VL2 uses a secondary or remote VS to establish telephone connections.

9.3.6.6.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.6.2 Related Topics

[Voice Server Groups and Remote Connections](#)

9.3.6.6.3 Configuration

To enable the VL to use a remote VS to establish telephone calls, simply insert the name of the remote VS into the registry key “TCVLink2\VoiceServerName” [REG_SZ]. The key is empty per default.

9.3.6.7 Answering Machine Detection

The VL2 has the ability to detect if the called party is an answering machine. Sometimes this feature is also called “mailbox detection”.

If the VL2 detects a mailbox, it can immediately disconnect and return an appropriate response or play an alternative prompt.

9.3.6.7.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

Only works with the LS1 as telephony hardware.

If a remote VS is used this remote VS must have at least KCS 7.66.01 installed.

9.3.6.7.2 Related Topics

[Answering Machine Detection](#)

9.3.6.7.3 Configuration

Enter the parameter string for the answering machine detection into the registry key "TCVLink2\Mailboxdetection" [REG_SZ]. The default value is "DetTime=60000". A description about the parameter string and possible values can be found in the chapter [Answering Machine Detection](#).

9.3.6.8 Individual IVR Plug-in

Per default, the VL2 uses VoiceXML as target for the LSR. In some circumstances, this might not be desired. Maybe because the alias is used by a different plug-in than VoiceXML or another plug-in than VoiceXML should handle the IVR-dialogs.

9.3.6.8.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.8.2 Related Topics

[Aliases](#)

[LSR – Local Session Redirection](#)

9.3.6.8.3 Configuration

Simply change the registry key "TCVLink2\VoiceXMLAlias" [REG_SZ] to the desired alias.

9.3.6.9 Address Syntax Customization

As the VL2 does not have a fixed set of send options like the VL had, a new way of defining send options was implemented. To allow maximum flexibility for the configuration of send options, the parsing was implemented in terms of regular expressions. If you are unfamiliar with regular expressions, please refer to [\[21\]](#)

There are two possible send option syntaxes. The old syntax, which was used by the original VL, and the new syntax. First the VL2 checks against the old syntax and if this fails the VL2 assumes that the new syntax is used.

The VL2 interprets the new send options as key – value pairs, leading to the only restriction that the regular expression must return, in case of a match, three results.

- 1.) The key. Which is also called the "tag".
- 2.) The value.

3.) The rest of the send options.

The VL2 itself uses only one send option, namely the language tag.

The name of the language tag can be configured separately so it is possible to address

"<<LANGUAGE=EN>>" instead of "<<LNG=EN>>"

9.3.6.9.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.9.2 Related Topics

[Customized Send Options](#)

9.3.6.9.3 Configuration

Attention: Changing this configuration may lead to unpredictable behavior and is done on own risk.

Note: The TC/VLATransformer uses the same regular expression. Be sure that any changes are done in both places (see [Customized Send Options](#))

The regular expression for the old syntax can be configured via the registry key "TCVLink2\SendoptionsRegex_Old" [REG_SZ] with the value "(.*)<([A-Za-z]{2,})?(-([01]?[01]))?>(.*)" as default.

The regular expression for the new syntax can be configured via the registry key "TCVLink2\SendoptionsRegex_New" [REG_SZ] with the value "[w\:\s]*<<([\w]+)=([\w]+)>>(.*)" as default.

9.3.6.10 TCSI Codepage Conversion

Whereas the playback of messages with a Middle Western character set used to work, the playback of messages with other codepages lead to unrecognizable results.

This was due to a missing conversion between of the TCOSS codepages.

Now, the VL2 is capable to use the TCSI codepage conversion.

9.3.6.10.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The TSCI Poller must be configured as poller

9.3.6.10.2 Configuration

Unfortunately the configuration must be done in several places.

The code page of the VS must be configured with the registry key "General\Codepage" [DWORD]. The default is "1252"

The TCOSS code page must be configured with the registry key "TCVLink2PolITCOSS\Codepage" [DWORD] and "VLATransform\Codepage" [DWORD]. The default for both keys is "0"

For details about how codepage conversion works please refer to the TC/Link manual.

9.3.6.11 Recipient Properties

During the message transformation, TCSI properties from the message header can be merged into resulting VoiceXML file ([Message Properties](#)). Additionally the VL2 will read the TCSI properties from the recipient entry (not the address list) and provide the content of these properties to VoiceXML over the “msgparam/plugindata” mechanism.

One possible recipient field would be for example “TS_FREETEXT”.

To access this field within the VoiceXML file, use the following code.

```
<log>
The value of TS_FREETEXT is
<value expr="com.topcall.PluginData.TS_FREETEXT.Index0" />
</log>
```

So the syntax is quite simple. It starts with “com.topcall.PluginData.” followed by the recipient property name like “TS_FREETEXT” and ends with “.Index0”. This “Index0” is required by the mapping mechanism and must not be omitted.

9.3.6.11.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The TSCI Poller must be configured as poller

9.3.6.11.2 Related Topics

[Message Properties](#)

9.3.6.11.3 Configuration

The registry key “TCVL2PolITCOSS\RecipientFields” [REG_MULTI_SZ] can hold the list of recipient properties where every property is written into a new line. This list is empty per default.

Possible values are e.g.: “TS_FREETEXT”, “TS_FULLSCREEN” and “TS_COMPANY”.

9.3.6.12 Poll Cycle

The VL2 contains a polling mechanism to check if new messages are available in the message store. Normally these messages are stored on TCOSS but might also be stored somewhere else. This pull mechanism has some advantages but also disadvantages when compared to a push mechanism. A typical push mechanism would be an SMS system where the message is “pushed” to a mobile phone instead of having the mobile phone “pull” it. The main advantage is that “pull” is simple to implement and does almost always work. The main disadvantage is that it is not “realtime” and that it adds overhead when there is actually nothing to do (e.g.: No message waiting).

The VL2 offers much flexibility in how much real-time is needed and how much overhead is caused.

The main parameter is the “poll cycle”. This is the time period the VL2 waits before the message store is asked again if a message is waiting. More exactly it is the time period the VL2 will wait after being told that there is no message waiting by the message store. This means that if the polling was successful and a message was retrieved, the poll cycle would not apply. In such a case, the VL2 will try to process the next message as fast as possible.

9.3.6.12.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.12.2 Related Topics

[VL2 - Polling](#)

[VL2 - Timeouts](#)

[Enhanced Message Termination](#)

9.3.6.12.3 Configuration

To configure the poll cycle, set the registry key “TCVLink2\PollCycle” [DWORD] to the desired duration in seconds.

To be honest, the poll cycle parameter is only a short cut to the time out with the time out code “101” (“No new message found” see [VL2 - Timeouts](#)).

9.3.6.13 Timeouts

Because of the way the VL2 manages timeouts, also other timeouts than timeouts for errors can be configured. Furthermore it is not always an error which requires a timeout, e.g.: the shut down time out. In this chapter each event requiring a timeout will be referred as error.

To minimize the configuration overhead, the VL2 has separated the error codes into different ranges.

Each range has its own default value, so it is not necessary to configure every ISDN Info code individually.

The VL2 uses a four level search to find a timeout.

- 1.) It searches for the timeout directly
- 2.) If the timeout is not found, the VL2 will search for the default timeout of the range the timeout is in.
- 3.) If this does not work, the error code 0 will be used as fallback.
- 4.) Last but not least, there is a hard coded time out of 10sec if the error code 0 is not configured.

9.3.6.13.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.13.2 Related Topics

[VL2 - Polling](#)

Enhanced Message Termination

9.3.6.13.3 Configuration

The timeout mapping is configured with the registry key “TCVLink2\Timeouts” [REG_MULTI_SZ].

Each line holds one mapping of the form “error code = timeout” E.g.:

```
101=10000
```

This would mean that if no message was found the poller should wait 10 seconds.

The following table is a list of error codes and their meaning.

Error code	Remark
0	Success, used when a message was found. The timeout value should be very small. E.g.: 10
1 – 15	TCRT errors. Please find a description of them in Enhanced Termination - Errors
100	Default for poller errors.
101	No new message found.
102	An error occurred during message retrieval.
200	Default for message related error which occurred in the VL2
201	The channel refused to play the message for some reason
202	There was no recipient specified
203	The message could not be processed because the VL2 was closing.
300	Default for misc. events
301	Used to determine how long the VL2 should wait for a poller to shut down.
302	Defines how long the VL2 should wait for the message response to be processed.
303	Defines how long a channel should wait for a new message. After this time elapses, the VL2 returns the control of the channel for 10msec. After this 10msec the channel will wait again for new messages.
305	Used during inactivity periods. Defines the interval in which the VL2 will check if the inactivity period is still active.
>4000	ISDN Info codes. These error codes are described in Enhanced Termination - Errors . These error codes are treated like normal TCRT errors.

Table 14 – VL2 Error Codes

It is of no use to configure timeouts for all possible error codes. E.g.: If the VS is really out of memory a timeout will not change much.

Please note that although it is possible to enter a timeout value of 0msec, this is not recommended.

9.3.6.14 Enhanced Message Termination

The Enhanced Message Termination (EMT) allows a more flexible and extensible way to configure how the VL2 should terminate a message.

9.3.6.14.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.14.2 Configuration

The configuration is done with an xml file. This xml file is usually called “Err2Break.xml” and is stored in “C:\TCOSS\System”. Name and path can be changed with the registry key “TCVL2PolITCOSS\Error2BreakFile” [REG_SZ].

Within the Error2Break (E2B) file, each row represents a rule.

The syntax of the E2B file is:

```
<err2break>
  <error>
    <err>0</err>
    <response>ACCEPTED</response>
    <code>GB</code>
    <note>Incident Accepted</note>
    <state>2698</state>
    <laststate>2698</laststate>
  </error>
</err2break>
```

It is possible to define as many `<error>` tags as necessary.

The `<response>` tag is optional.

The tags have the following meaning:

- `<err2break>` is the root tag. The tag must only exist exactly once.
- `<error>` this tag specifies one entry.
- `<err>` This is the error code supplied with the response to the message. (0 = no error occurred)
- `<response>` This tag is optional and only necessary if the error code is not unique enough. (ACCEPTED = This entry will be used if the supplied response is “ACCEPTED”)
- `<code>`This code will be translated into the error string in TCfW (GB = TCfW will display “The message was accepted”). A list of these codes can be found in the subchapter [Action](#).
- `<note>` A max 24-characters-long string that is shown in TCfW in the column “response”
- `<state>` The break code to use if there are retries left. (2698 = Set to “Sent OK”)
- `<laststate>` The break code to use if there are no retries left. (2698 = “Sent OK”)

It is possible to have multiple `<error>` tags with the same `<err>`. To distinguish them, the `<response>` tag is used. This is necessary because if the recipient accepted or rejected the call, no error occurred but the result must be treated differently.

E.g.:

```
<err2break>
  <error>
    <err>0</err>
    <response>ACCEPTED</response>
    <code>GB</code>
    <note>Incident Accepted</note>
    <state>2698</state>
    <laststate>2698</laststate>
  </error>
  <error>
    <err>0</err>
    <response>ACCEPTED</response>
    <code>G0</code>
    <note>Incident Rejected</note>
    <state>430</state>
    <laststate>430</laststate>
```

```
</error>
</err2break>
```

If a response shows an error code which is not defined in the E2B, the TCIS-Poller will use the “*Default Poller errors*” Code (Table 14 – VL2 Error Codes).

9.3.6.14.3 Errors

Because there are a lot of possible errors and most are very unlikely, the following table shows only the most common of them. If one of the unlikely errors occurs, it will be caught by the default rule.

Name	Meaning
DEFAULT	Applies if the closing state has no rule for the specific error
0	Ok
1	TCRT: ErrInt
2	TCRT: OutOfMemory
3	TCRT: BadParam
4	TCRT: Timeout
5	TCRT: NotSupported
6	TCRT: NotConnected
7	TCRT: BadChannel
8	TCRT: BadChannelType
9	TCRT: BadEngine
10	TCRT: Busy
11	TCRT: NoResponse
12	TCRT: Err
13	TCRT: InvalidHandle
14	TCRT: Cancelled
15	TCRT: ServerNotFound
2000	Address parsing error
4097	ISDN Info 1
4296	ISDN Info 200
4298	ISDN Info 202

Table 15 – TCRT Error Codes

There are 120 ISDN Info codes and some additional TC specific ISDN codes. The ISDN info codes are calculated by a fixed value 4096 plus their corresponding ISDN info code. E.g.: ISDN Info 1 = 4096 + 1 = 4097

Some additional error codes can be found in [VL2 Timeouts](#)

For more information about the different errors and their meaning, read the TCRT-Manual or the TCOSS ISDN manual.

9.3.6.14.4 Break Codes

The following table holds the most common break codes.

Name	Value
Break 0	310
Break 1	320
Break 2	330
Break 3	340
Break 4	350
Break 5	360
Cancelled	400
End Of Retries	420
Rejected	430
Sent Ok	2698

Table 16 – TCSI Break Codes

More information about break codes can be found in the TCSI Manual.

The behavior of TCOSS can be configured via the file “ATAMCONFATC” in the “+TECH” folder. More information about the configuration of the “ATAMCONFATC” can be found in [\[22\]](#).

9.3.6.14.5 Action

The 2 digit code is used by client programs to display a correct error description. A list of all possible values can be found in the file “TC01.err”, “TC02.err” (language specific) under “C:\topcall\shared”

Here is a list of actions used by the TC/VoiceLink.

Code	Description
G0	The message has been rejected
G1	The delivery of the message has been refused
G2	Line busy or incorrect number
G3	The recipient did not response to the questions
G4	The message was delivered, but the reception has not been confirmed
G5	General internal error
G6	Internal error during playback
G7	Internal Error (wrong error in wrong state)
G8	The delivery of the message has been interrupted by the recipient
G9	Error while preparing the message. Please check send options and content
GA	Unable to establish call. Please check if the telephone system is working
GB	The message was accepted
GC	ISDN Error please check your telephone system.
GD	The priority of the message is invalid for the current time.
GE	The call was either not answered or it was rejected.
GN	The message has been recorded to the voice box.

Table 17 – VL Action Abbreviations

9.3.6.15 Disable TTS

Unlike for the VL, it is not possible to disable the TTS explicitly for the VL2. It is possible to write VoiceXML files which rely only on pre-recorded audio files. But for a VL2 which never plays TTS, also the error message and other default messages must be pre-recorded.

9.3.6.15.1 Configuration

None

9.3.6.16 Tracing

The VL2 can do quite heavy tracing and therefore it should be only activated for diagnostic reasons. Otherwise the performance of the VS can drop significantly.

Every line traced by the VL2 starts with “TCVLINK2:”.

9.3.6.16.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.16.2 Configuration

The different trace levels can be set with the registry key

-) “TCVLink2\Tracelevel” [DWORD] for the VL2 .
-) “TCVL2PolITCOSS\TraceLevel” [DWORD] for the TSCI poller
-) “VLATransform\TraceLevel” [DWORD] for the TC/VLATransformer
-) “TCVL2LogTOCSS\TraceLevel” [DWORD] for the tcsi – logger
-) “TCVL2LogFile\TraceLevel” [DWORD] for the file logger.

The default for all the trace levels is “0”

Trace level (decimal)	Meaning
0	Trace only errors
1	Trace errors and warnings
3	Everything from above and general purpose processing information
7	Everything from above and function calls.
15	Everything from above and debug information

Table 18 – VL2 Trace Levels

9.3.6.17 Directories Used

The TSCI poller requires a temporary directory where the converted VoiceXML files are stored.

A second directory is the templates.

9.3.6.17.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.17.2 Related Topics

[Templates](#)

[Keep Remainings](#)

[Default Template](#)

9.3.6.17.3 Configuration

The name of the template directory is stored in the key “VLATransform\TemplateDir” [REG_SZ] with the default value “C:\TCOSS\System\VLA\Templates”.

The temporary directory for the VoiceXML conversion is configured with the registry keys “VLATransform\OutDir” [REG_SZ] and “TCVL2PolITCOSS\ConversionPath” [REG_SZ]. Both keys **MUST** point to the same directory. The default value is “C:\TCOSS\SYSTEM\VLA\OutDir”.

9.3.6.18 Base URL

As previous versions of the VL2 supported a more distributed scenario there was a feature called “Base URL”. With the current version this not needed anymore but for compatibility reasons the configuration possibility is still there.

9.3.6.18.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.18.2 Configuration

The name of the base URL is stored in the key “VLATransform\BaseURL” [REG_SZ], which is empty by default and **MUST** be empty if used with the TC/VoiceLink 2.2.

9.3.6.19 Old VoiceLink Compatibility

The TSCI poller is responsible for the backward compatibility to the “old” VL. To ensure this, next to the new addressing syntax, also the old one is still supported.

If the TSCI poller detects a message with the old addressing syntax, a special template will be loaded. This template emulates the behavior of the old VL.

9.3.6.19.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.19.2 Related Topics

[Default Template](#)

[From VL to VL2](#)

9.3.6.19.3 Configuration

The name of the template which emulates the old behavior is configured with the key “VLATransform\OldTemplate” [REG_SZ] and holds the value “OldVoiceLink.vxml” as a default.

Note: The templates specified in the registry may have any file extension.

If the TSCI poller does not detect send options, the key “VLATransform\NolsOld” [DWORD] will specify which template to use. If set to “1”, the template specified in “VLATransform\OldTempate” will be used. If set to “0”, which is the default, the template specified in “VLATransform\DefaultTemplate” will be used.

To manipulate the behavior of the old VL template, when no send options are present the following registry key can be used:

“VLATransform\Menu1” [DWORD] activates or deactivates both the first and the second menu. The first menu asks the recipient if he wants to hear the message at all. The second menu asks the recipient to accept or reject the message.

Note: The registry key “VLATransform\Menu2” [DWORD] is not used.

The default language is configured via the “xml:lang” tag of the template.

9.3.6.20 Default Template

If a message is transformed and there is no template explicitly specified, or that there are no send options at all, the TSCI poller will use a default template.

9.3.6.20.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.20.2 Related Topics

[Old VoiceLink Compatibility](#)

9.3.6.20.3 Configuration

The name of the default template will be stored in the key “VLATransform\DefaultTemplate” [REG_SZ]. The default is “OldVoiceLink.vxml”.

To determine which template to use if no send options are specified, the key “VLATransform\NolsOld” [DWORD] is consolidated. If set to “1”, the TSCI poller will load the template specified in “VLATransform\OldTemplate”. If set to “0”, the TSCI poller will load the template specified in “VLATransform\DefaultTemplate”.

Maybe you have noticed that this is the same value as for the “VLATransform\OldTemplate” [REG_SZ] key. Keep this in mind when changing the value of the “VLATransform\DefaultTemplate” key. It might be necessary to change the “VLATransform\NolsOld” key as well.

9.3.6.21 Message Properties

The TSCI poller offers the possibility to replace the placeholders in a template by the content of message property. Such a property might be the subject, the priority or a specific text block from the body.

The only drawback is that the TSCI poller does only use properties which are specific for the message and not, for example, for the recipient. This means that the TSCI poller will not replace a placeholder with the full name of a recipient. This is done by the VL2.

Only fields from the header or the transfer envelope can be configured to be used. The fields from the “message body” like the body text are used automatically.

To locate the field to use, the TSCI poller uses the TSCI Path to the field. Because this is error prone the TSCI poller uses a short cut for the header and transfer envelope.

For example, to tell the TSCI poller to read the subject from the message header, the line "HEADER.TS_REF" must be added to the configuration. To get the originator information, add the line "MSG.TS_ORIGINATOR_INFO" to the configuration.

Additionally, the TSCI poller knows some pre-defined TCSI properties:

- \$TCVLINK2.Menu1\$ – Holds the value for the menu1 of the old VL template
- \$TCVLINK2.Menu2\$ – Holds the value for the menu2 of the old VL template
- \$body.text.X\$ - Holds the content of a text block at the Xth TCSI index of the block in the message.
- \$attachment.filename.X\$ – Holds the filename of the attachment at the TCSI index X.
- \$attachment.text.X\$ – Holds the alternative content of a binary block or an image block.
- \$body.text\$ – Resembles the list of body.text.X and attachment.filename.X in the order they appear in the message. If a message contains a "body.text.0", an "attachment.filename.1" and a "body.text.2" the body.text will resolve to "\$body.text.0\$ \$attachment.filename.1\$ \$body.text.2\$"

9.3.6.21.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.21.2 Related Topics

[VL2 - Templates](#)

[Recipient Properties](#)

[Placeholder Identification](#)

9.3.6.21.3 Configuration

The list of fields can be configured by using the key "VLATransform\TCSIFields" [MULTI_REG_SZ] where each line holds a field. The default is "HEADER.INT_PRIORITY_TO HEADER.TS_REF MSG.TS_ORIGINATOR_INFO MSG.TIME_ACTION".

9.3.6.22 Keep Remainings

This feature is mainly for troubleshooting purposes. Normally, the TSCI poller will delete the temporary directory which was created in the Out-Dir if the message was not successfully transformed.

It is recommended to deactivate this behavior if the TSCI poller shows warnings that the resulting VoiceXML file is not valid.

9.3.6.22.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.22.2 Related Topics

[Directories Used](#)

9.3.6.22.3 Configuration

Set the key “VLATransform\KeepRemainings” [DWORD] to “1” if you want the TSCI poller to not remove obsolete files. Set the key to “0” to have the normal behavior again.

9.3.6.23 Placeholder Identification

To search for a placeholder in a template, the TSCI poller uses regular expressions (See [\[21\]](#)). Unlike as for the send options, this regular expression cannot be fully configured. Only the pre- and suffix of a placeholder are configurable. This prefix and suffix are also called “token”.

The regular expression for the search for placeholders is:

```
([\w\.\-\]+)
```

With the default token the final regular expression is:

```
\$([\w\.\-\+])\$
```

This means that a placeholder must start with a “\$” and end with a “\$”. In between the two “\$”, at least one alphanumerical character, a “.” or a “-” must be present.

E.g.:

- \$a.c\$ = Valid
- \$\$ = Invalid
- \$a\$ = Valid
- \$ac\dc\$ = Invalid (“\” is no alphanumerical character)

9.3.6.23.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.23.2 Related Topics

[Customized Send Options](#)

9.3.6.23.3 Configuration

The token is configured with the registry key “VLATransform\ReplacerToken” [REG_SZ] and is set to “\\$” per default.

Note: If you need to change the token, do not forget to escape it if necessary.

9.3.6.24 Customized Send Options

The TSCI poller knows three different send option styles.

- First, the old style used by the VL.
- Secondly, there is a now more extendable style used by the VL2.
- The third style will apply if no send options are specified at all.

Whereas the third type does not require any configuration besides compatibility issues (see [Old VoiceLink Compatibility](#)), the first and second type can be complex.

It is essential for the TSCI poller to determine the correct send options style, because based on the style a specific template is loaded.

The old style looked like:

```
[<[EN-] [ [0]1]>]123
```

EN ... the optional language identifier

0 1 ... optionally specified which menu was active or not.

The new send option style is more generic and was designed primarily with automation in mind.

```
[<<TAG=VALUE>>] [<<TAG=VALUE>>] 123
```

TAG ... is the name of a specific option. E.g.: LNG for the language option.

VALUE ... is the value of the tag. E.g.: If the tag is "LNG" the value might be "EN" for English.

To find the current send option styles the TSCI poller first checks if the send options match the old style. If this is not successful, the TSCI poller will check the new send options style.

While the look of the send options was already highly configurable with the old VL, the TSCI poller offers much more freedom concerning the send option layout.

This freedom comes from the fact that the TSCI poller uses regular expression to parse the send options. While normally only the separators between the options were configurable, now the layout possibilities of send options are nearly unlimited.

The default regular expression for the old style is:

```
(.*<(([A-Z]{2,})?(-([01]?[01]))?)>(.*)
```

The default regular expression for the new style is:

```
[\w\:] * << ([\w]+) = ([\w]+) >> (.*)
```

Basically, for each style the TSCI poller provides only a few guidelines which must be fulfilled.

For the old style, these guidelines are:

- The regular expression must return 5 groups.
- The first group contains the prefix (Everything before the "<"). This group is ignored but must exist.
- The second group contains the language identifier which might be empty.
- The third group contains the separator between the language identifier and the menu options. This group is ignored but must exist.
- The forth group contains the menu options. The menu options must be one of the following strings: "", "0", "1", "00", "01", "10" and "11".
- The fifth group holds the recipient's address.

The new style has fewer rules:

- The regular expression must return 3 groups.
- The first group contains the tag which must not be empty.
- The second group contains the value for the tag which can be empty.
- The third group holds the rest of the address including the remaining send options.

The regular expression is reapplied to the third group as long as send options are found.

If the TSCI poller detects the new send option style, it will search for the following tags:

- A language tag. This tag specifies the language to use for the recipient. Per default “LNG”. The value of the tag can be any string.
- A template tag. This tag specifies the template to use for the message. Per default “TMPL”. The value of the tag must be the name of a template without the file extension. The template must be located in the template directory. The TSCI poller automatically adds the extension “.templ” to the value of the tag.

Note that some tags are taken from the originators send options; others are taken from the recipient’s send options.

- From the originator the template tag is used.
- From the recipient only the language tag is used.

9.3.6.24.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

9.3.6.24.2 Related Topics

[Old VoiceLink Compatibility](#)

[Multi-lingual](#)

[Directories Used](#)

[Address Syntax Customization](#)

9.3.6.24.3 Configuration

The regular expression for the old style send options is configured by the registry key “VLATransform\SendOptions1” [REG_SZ]. The default is “(.*)<([A-Z]{2,})?(-([01]?[01]))?>(.*)”.

The regular expression for the new style send options is configured by the registry key “VLATransform\SendOptions2” [REG_SZ]. The default is “[\\w:]*<<([\\w]+)=([\\w]+)>>(.*)”.

The name of the language tag can be configured by the registry key “VLATransform\LanguageTag” [REG_SZ]. The default is “LNG”.

The name of the template tag can be configured by the registry key “VLATransform\TemplateTag” [REG_SZ]. The default is “TMPL”.

Note: Make sure that the configured regular expression are the same the VL2 uses ([Address Syntax Customization](#))

9.3.6.25 Number Locking

Contrary to an email, it is not recommended to send more than one VL2 message simultaneously to one recipient as he/she can only handle one message at a time. To prevent the VL2 from sending more than one message to the same phone number, the feature “Number Locking” must be activated.

9.3.6.25.1 Prerequisites and Additional Installation

TCOSS 7.69.04

TCSI 2.62.00

TCfW 5.24.00

9.3.6.25.2 Configuration

The number locking can be configured on a per-queue base. To activate it, check the check box next to "Number locking" as seen in Figure 75.

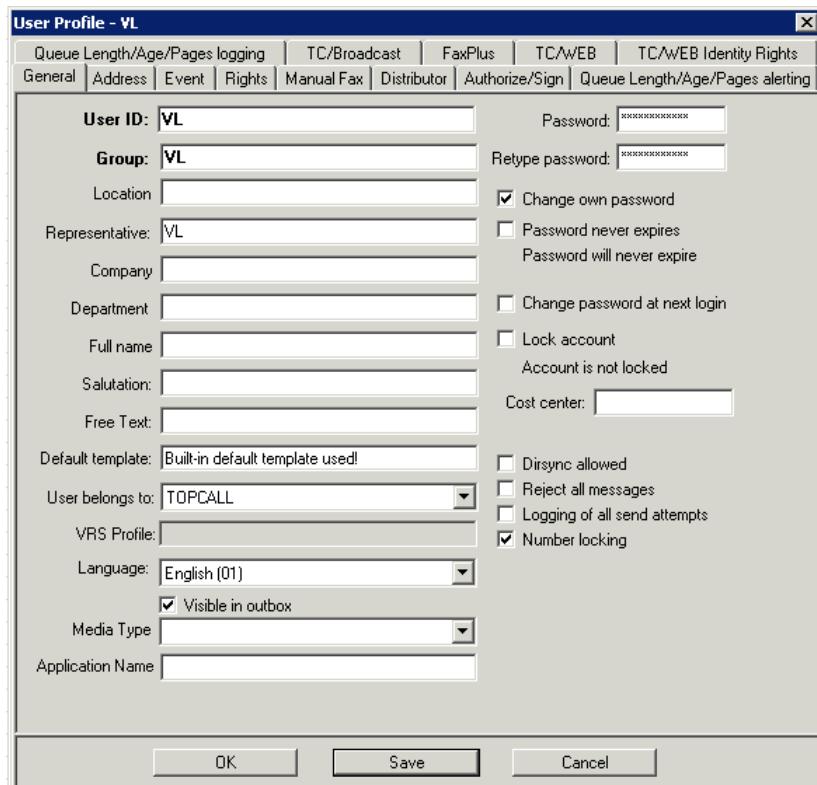


Figure 75 – TCfW Number Locking

It is only necessary to activate the number locking in the "VL" queue because the actual calls are triggered by this queue.

9.3.6.26 Text2Wave Pre-Conversion

As described in the Concepts and Architectures chapter [Text2Wav Pre-Conversion](#), the VL2 is capable of converting parts of the VoiceXML into audio files. This has the advantage that the VL2 does not need to do this anymore during the call which leads to a better response time.

9.3.6.26.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The TSCI poller must be used.

9.3.6.26.2 Related Topics

[Concepts and Architecture: Text2Wav Pre-Conversion](#)

9.3.6.26.3 Configuration

This feature is only available if the TSCI poller is used. The pre-conversion conversion can be activated or deactivated with the key “..\\VLATransform\\Convert2Wave” [DWORD]. It is deactivated (“0”) by default. To activate it, set the key to “1”.

9.3.6.27 Customizing Audio Extensions

As the VL2 is not only capable of playing text but can play audio files, it is necessary to tell the VL2 which attachment of the message should be used for playback. The VL2 uses a list of extensions which is compared to the extension of the attachment. If the extensions do match, the attachment is embedded into template within an <audio> tag.

E.g.: The message contains an attachment called “something.wav”

```
<audio src="file:///c:/tcoss/system/vla/outdir/a-very-long-guid/something.wav">
something.wav
</audio>
```

This offers the possibility that if a file can't be played, because it is no audio file after all, at least the file name is mentioned.

9.3.6.27.1 Prerequisites and Additional Installation

TCSP 7.80.04 or higher

The TSCI poller must be used.

9.3.6.27.2 Configuration

This feature is only available if the TSCI poller is used.

The list of extensions can be added to the registry key “..\\VLATransform\\AudioExtensions” [REG_MULTI_SZ]. The default value is “.wav” and “.tcs”. It is important that the extension does start with “.” and do not have any trailing blanks. The extensions are case insensitive.

9.3.7 TC/VRedirect

9.3.7.1 Caching Of CustomerIDs

The main functionality of TC/VRedirect is to redirect calls based on their destination number to a specific VS. This is done by querying a TCOSS media server for the customer ID which is associated with the destination number. To speed up this query, TC/VRedirect caches the results of the query.

9.3.7.1.1 Configuration

The registry key “TCVRedirect\\CacheSize” [DWORD] specifies the maximum cache size. The default is 1000. The maximum is 2147483647.

With the registry key “TCVRedirect\CacheTTL” [DWORD] it is possible to control how long a unused cache is kept in cache. The time unit is seconds and the default is 60. The maximum value is 2147483647 (~70 Years).

9.3.7.2 Service of a Voice Call

To look up the customer id related to the destination number, TC/VRedirect needs to know the TCOSS service which is used for voice address.

9.3.7.2.1 Configuration

The name of the service associated with a voice call can be configured with the registry key “TCVRedirect\ServiceName” [REG_SZ]. The default is “Voice”.

9.3.7.3 Tracing

TC/VRedirect is also capable of tracing.

9.3.7.3.1 Configuration

The registry key “...\\TCVREDIRECT\\TraceLevel” [DWORD] is a bit field where different trace levels can be combined. Available trace levels are:

Name	Value	Comment
ERROR	0x00	Is always traced
WARNING	0x01	Traces warnings
INFO	0x02	Traces additional information. The most interesting part is that when the VS is shut down, a statistic will be traced about cache size, hits and misses.
FUNCTION	0x04	Traces function calls
DATA	0x08	Additional data is traced. (Open TCSI objects)

Table 19 – TC/VRedirect Trace Levels

10. Scenarios and Tutorials

10.1 TC-Prod-Voice – Typical Installation

10.1.1 The Script

A typical company uses the VS primarily as mailbox. Because of the need to be cutting edge the attendant and TC/PreProcessor with MetaMail support are configured. Additionally a VL can be used for outbound calls and the users are able to use the TC/Player to playback wav-files via the VS.

In this scenario, the VA receives ~60 calls per day. ~97% of the calls use the VA to leave a message or are at least forwarded to the VA so they can leave a message. ~3% of the calls to VA are done to access the mailbox and listen to voice mails. Only 1% of all these calls involve a call transfer of any type.

The VL sends ~27 messages a month. The most common use is to send 3 – 6 messages at once with the VL.

10.1.2 The Actors

As hardware:

- A computer (2,8 GHz, 1GB Ram and 60GB HDD)

As software:

- VoiceAccess with an enabled Attendant
- TC/PreProcessor with Metamail support
- SAPI 5
- Acapela v5.200
- SPI-LN with Lotus Notes 7.0
- 1 instance of the VoiceLink
- TC/Switch for TC/Player

10.1.3 The Setting

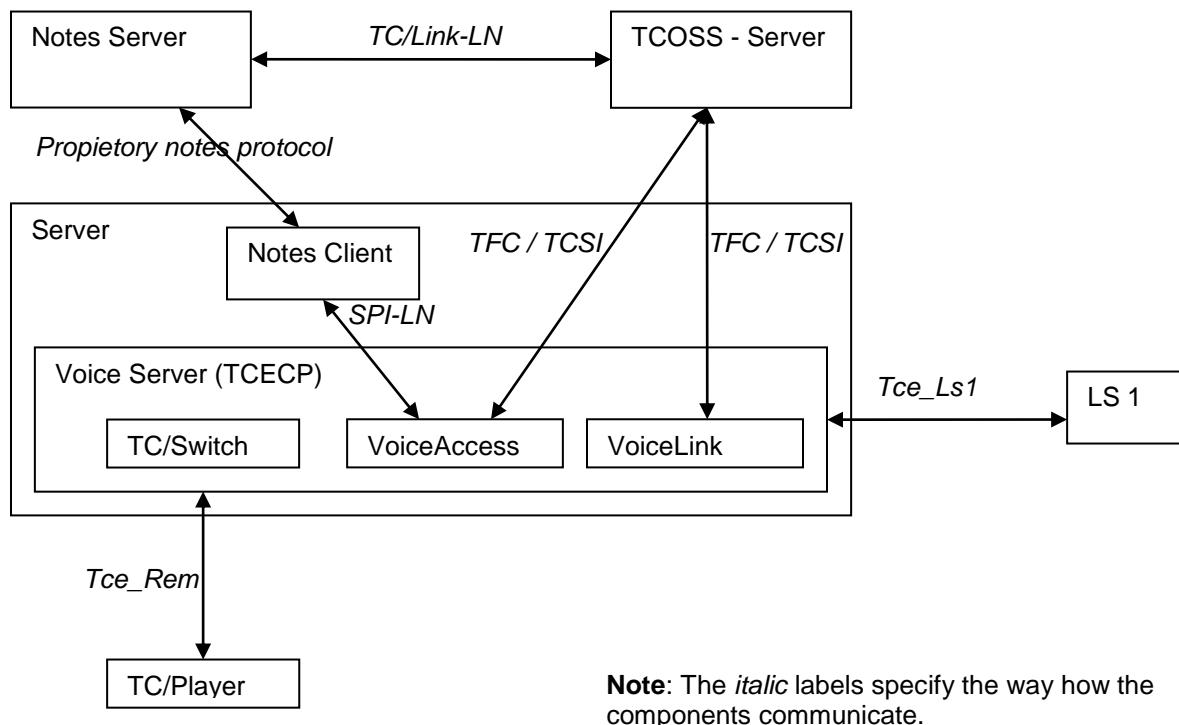


Figure 76 – A Typical Scenario

10.2 The xSP – Service Provider

10.2.1 The Script

A service provider would like to host many VS in a cheap as possible way. The easiest way to save costs is to run as many VS on one physical server as possible. The goal is to run 20 parallel VS instances on one server.

To decrease maintenance the xSP only uses VA with attendant. (Service customers can play their voice message via TC/Player on their telephone)

As a tribute to the required performance, the used TTS engine is only the MS TTS engine.

10.2.2 The Actors

As hardware:

- 1 * Routing Server = Single Processor 1,4GHz, 512MB Ram
- N * Voice Server = Dual Processor 2,8GHz , Raid 1, 2GB Ram

As software:

- TC/VRedirect on the routing server.
- VoiceAccess with an enabled Attendant

- SAPI 5
- Microsoft TTS
- TC/Player – TC/Switch

10.2.3 The Setting

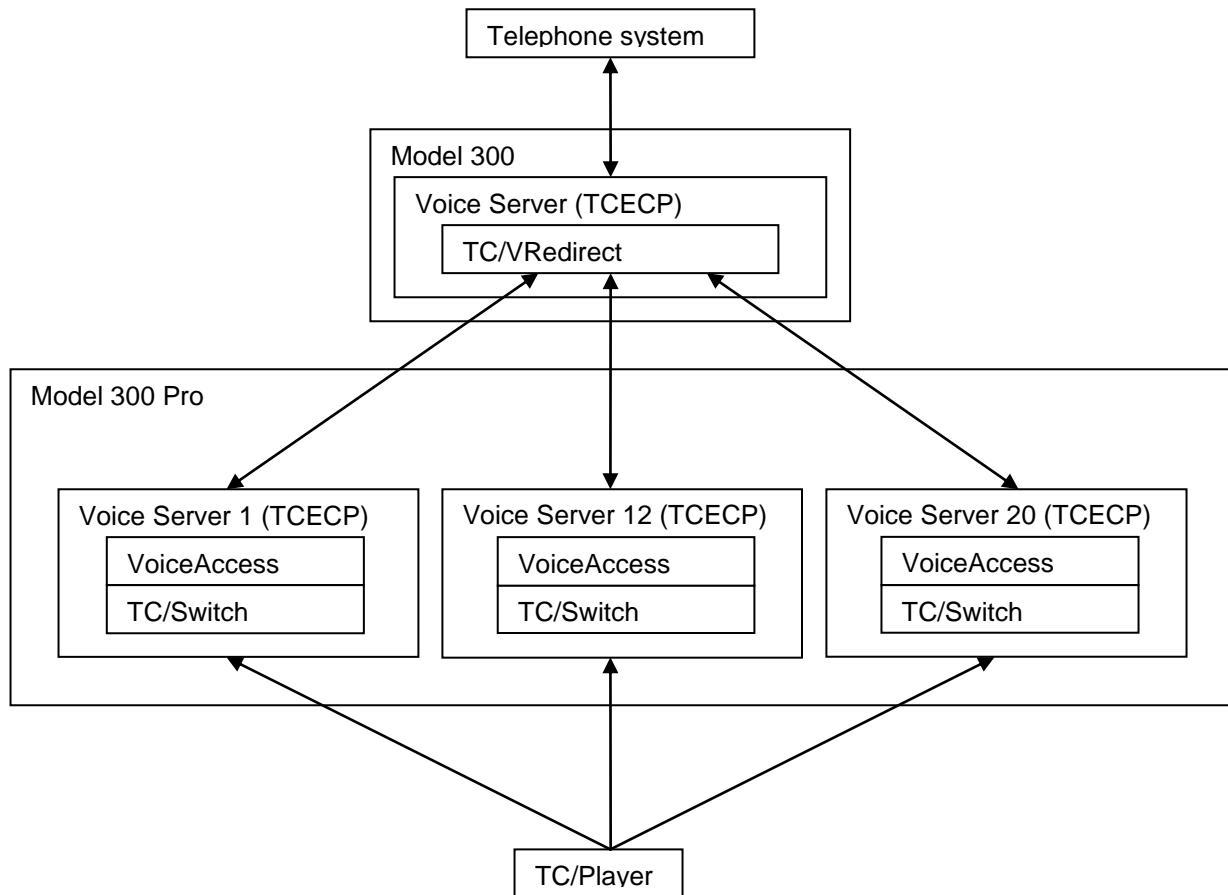


Figure 77 – Voice xSP Scenario Setting

Every incoming call is first processed by TC/VRedirect. Based on the called number, TC/VRedirect can compute the customer who should receive the call. With the help of the [VSG](#), TC/VRedirect forwards the call to the responsible VS.

The TC/Player on the other hand is only allowed to connect to VS in his configured [VSG](#). The VS will then establish the required connection to the telephone system over the routing server.

The only caveat in this scenario is that no SPIs can be used because of their limitations.

Detailed information about a VoiceXSP setup can be found in [\[10\]](#)

10.3 VoiceXML

10.3.1 The Script

An agency would like to offer a special service to their customers. A customer should have the possibility to extend his fishing license issued by the agency. The agency stores the data for the licenses in a database and already has a web application accessing this database.

For this scenario a simple VS with VoiceXML are sufficient. The web application has to be extended to response to requests from the VS with valid VoiceXML.

10.3.2 The Actors

As hardware:

- A computer (2,8 GHz, 1GB Ram and 60GB HDD)

As software:

- TC/VoiceXML
- SAPI 5
- Microsoft TTS

10.3.3 The Setting

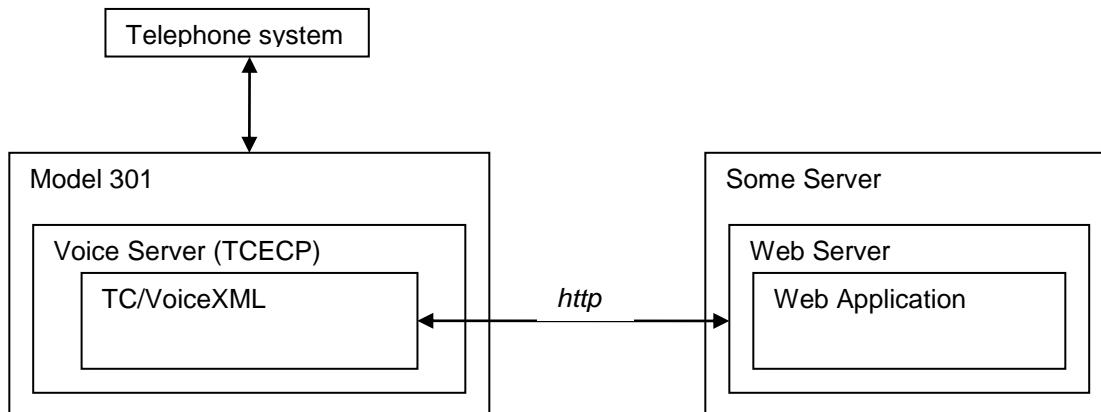


Figure 78 – VoiceXML Scenario Setting

10.4 TC/VoiceLink2

10.4.1 Customizing the TSCI Poller's Termination Behavior

If a message is sent, there are many possible error and success situations.

The VL2 tries to resemble the behavior of the VL as close as possible.

As already shown in [Enhanced Message Termination](#) the new format of the E2B uses XML instead of a comma-separated list.

To customize the behavior how the TSCI Poller terminates messages, this E2B file has to be edited.

The most important rule to remember is that the TSCI Poller always tries to find the entry which best matches the error code and response string.

E.g.: The error code “0” was issued and the response was “HANG UP”. First, the TSCI poller searches for an exact match. If this exact match is not found, an entry with the error code 0 but without response will be searched. If even this entry is not found, the TSCI poller will look for an entry with the configured default error code.

The shipped E2B can be used as a sample but may not contain all the response and error code combinations required for a certain situation. In order to find which response and error code combination were returned by the VL2, the trace level of the VL2 has to be set to 0x08. If this is done the VL2 will trace for every message the following three lines:

```
XXXXXX TCVLink2: MsgID: 12345
XXXXXX TCVLink2: Response: Some response
XXXXXX TCVLink2: Error code: 0
```

With this information new entries in the E2B can be created.

There are error situations which are not directly related to the recipient. If one of them occurs, it will not be a problem to retry sending the message.

10.5 TC/VoiceLink2 – Transformation of a Message

This chapter describes how the VL2 transforms different messages and where attention has to be paid when a template is to be written.

10.5.1 Writing a Template

When writing a template, there is only one important rule:

At the end of a transformation, a valid VoiceXML must be generated.

In some cases, the template might not be a valid XML document at all. But in the end, when all transformations have been done, the result must be a valid VoiceXML file.

The second most important rule is:

Every placeholder must be resolved.

If a template has a placeholder which the VL2 cannot replace because no matching property was found, the transformation will be aborted.

A template that only plays the content of a message to the recipient could look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/vxml
      http://www.w3.org/TR/voicexml20/vxml.xsd">

    <form>
      <block>
        <prompt>
          $body.text$
```

```

        </prompt>
    </block>
</form>
</vxml>
```

Now assume that a message contains the following blocks (the --- mark the beginning and the end of a block):

```

Hi!
How are you. I send you this file for review.
---
<<Attachment review.doc>>
---
Krgds
Me
```

The resulting VoiceXML would look like:

```

<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/vxml
      http://www.w3.org/TR/voicexml20/vxml.xsd">

    <form>
        <block>
            <prompt>
                Hi!
                How are you. I send you this file for review.
                Review.doc
                Krgds
                Me
            </prompt>
        </block>
    </form>
</vxml>
```

The next example is a little bit more complex but still does only play the message to the recipient:

```

<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/vxml
      http://www.w3.org/TR/voicexml20/vxml.xsd">

    <form>
        <block>
            <prompt>
                You received a message with the subject:
                $HEADER.TS_REF$
                The originator wrote:
                $body.text.0$
            </prompt>
        </block>
    </form>
</vxml>
```

With the message:

```

Subject: Jazz Concert
---
Hi!
Would you like to go to the jazz concert next Saturday?
Krgds
Me
```

```
---
Ps.: Heres the price list.
---
<<Attachment prices.doc>>
```

The template would be resolved to:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/vxml
      http://www.w3.org/TR/voicexml20/vxml.xsd">

    <form>
        <block>
            <prompt>
                You received a message with the subject:
                Jazz Concert
                The originator wrote:
                Hi! Would you like to go to the jazz concert next Saturday?
                Krgds
                Me
            </prompt>
        </block>
    </form>
</vxml>
```

As you might have noticed, not the whole message is part of the resulting VoiceXML.

This is because only \$body.text.0\$ was defined.

The last example shows a possible template which is not a valid VoiceXML file per se.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/vxml
      http://www.w3.org/TR/voicexml20/vxml.xsd">

    <form>
        $body.text.0$
        <block>
            <prompt>
                <audio $body.text.1$ />
            </prompt>
        </block>
    </form>
</vxml>
```

This template relies heavily on the message to complete it.

The message:

```
Hi! How are you.
Krgds
Me
```

will not produce valid results.

A message for the above template must have at least two text blocks Where each contains some VoiceXML markup.

```
<block>
<prompt>
Hi! How are you.
```

```
Krgds
Me
</prompt>
</block>
---
src="http://somewhere.com/audio.wav"
```

Would produce a valid result.

10.5.2 The URL Message

This is the simplest message type the VL2 can receive.

A URL message contains only a URL of type [file://](#), [http://](#), or [https://](#) in the first text block.

If the VL2 receives such a message, the text block will simply be copied to a new message to the VL2. Nothing will be transformed or converted.

A sample message could look like this:

```
FROM: Me
TO: You
Subject: VoiceMessage 123
---
http://webserver.com/vxml/index.vxml
---
```

10.5.3 The VoiceXML Message

The reason why this message is named VoiceXML message is because the first text block is a valid VoiceXML document.

If the VL2 receives such a message the text block will be copied into a “index.vxml” file in a temporary directory.

A sample message could look like this:

```
FROM: Me
TO: You
Subject: Mozart
---
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/vxml
      http://www.w3.org/TR/voicexml20/vxml.xsd">

    <form>
        <block>
            <prompt>
                This year is Mozarts year.
            </prompt>
        </block>
    </form>
</vxml>
```

Firstly, the VL2 will check if the VoiceXML is valid. Then a temporary directory in the “OutDir” is created (e.g.: “C:\TCOSS\SYSTEM\VLA\OutDir\0815-ABCD-4711”)

The next step is that the VoiceXML is written to the “index.vxml” in the temporary directory (e.g.: “C:\TCOSS\SYSTEM\VLA\OutDir\0815-ABCD-4711\index.vxml”)

10.5.4 The Text Message

This is the most common form of a message. The VL2 will assume that a message is a text message if the text does not start with "<?xml" or with an URL.

If the VL2 receives such a message, the message will be analyzed and the TCSI properties will be read.

After that, the VL2 loads the specified template and starts to replace the placeholders with the TCSI properties.

If the message is:

```
Hi!  
How are you.  
Krgds  
Me
```

and the template file contains:

```
<?xml version="1.0" encoding="UTF-8"?>  
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"  
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
      xsi:schemaLocation="http://www.w3.org/2001/vxml  
                        http://www.w3.org/TR/voicexml20/vxml.xsd">  
  
    <form>  
      <block>  
        <prompt>  
          $body.text$  
        </prompt>  
      </block>  
    </form>  
</vxml>
```

The result would look like:

```
<?xml version="1.0" encoding="UTF-8"?>  
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"  
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
      xsi:schemaLocation="http://www.w3.org/2001/vxml  
                        http://www.w3.org/TR/voicexml20/vxml.xsd">  
  
    <form>  
      <block>  
        <prompt>  
          Hi!  
          How are you.  
          Krgds  
          Me  
        </prompt>  
      </block>  
    </form>  
</vxml>
```

This result is now converted and stored in the temporary directory. The workflow is the same as described in [The VoiceXML message](#).

10.5.5 The User Properties Message

This is the most sophisticated message type containing the user property markup which can be used to define additional properties to the TCSI properties. See [User Properties](#).

Beside the fact that the body text is actually some form of XML, it is treated like a normal text message.

This means that a template is loaded and the placeholders get resolved.

10.6 TC/SPI-LN Manual Installation

- Install Lotus Notes Client and be sure that the environment variable PATH is set to the Notes directory!
- Copy the file LCPPN412.DLL to the C:\TOPCALL\SHARED directory. Copy the TCSPILN.DLL and TFC.DLL to the directory C:\TOPCALL\SHARED\ and register these two files with regsvr32.
- Copy the TCPWD.DLL to the Notes directory (not in Notes\data – only Notes directory) and enter the following line after the section name [Notes] of the notes.ini:
`EXTMGR_ADDINS=TCPWD`
- Then open the registry editor and create the key "...\\SPI\\Notes\\DefaultNotesServer" [REG_SZ] and enter the name of the Notes Server (for example: ServerName/CertName).
- Also create the registry key "NotesAddin" [REG_SZ] and enter the password for the user ID. After the first start of the Notes Client or TCECP, the registry key will be encrypted.

10.7 Testing IVR without A Telephone System

Actually this is very easy if the VS has a soundcard installed. Change the registry key "...\\TCRT\\ShowEngineDialog" [DWORD] to "1". After a restart of the VS, a dialog should appear that looks like Figure 79 – Phone Emulator Dialog.

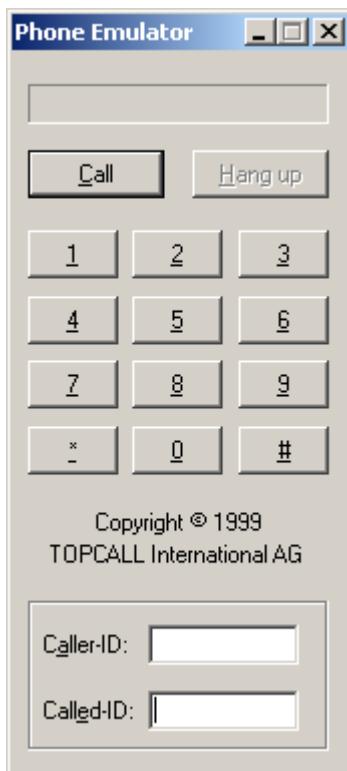


Figure 79 – Phone Emulator Dialog

This dialog emulates a phone and can be used to place calls to the VS.

In the field “Caller-ID”, enter the phone number of the caller.

In the field “Called-ID”, enter the phone number you would like the phone emulator to dial.

Then click Call

Note: It is **not** possible to receive calls from the VS with the phone emulator!

10.8 Removing the Old TC/VoiceLink

This chapter describes how to remove the VL components from the VS. It is not a tutorial about how to remove the VS.

- 1) Make sure the VS is not running.
- 2) In the registry, delete the key “..\TOPCALL\TCECP\TCVLink”.
- 3) Check if the registry key “..\TOPCALL\TCECP\General\Plugins” does not contain the string “vlink=1#TCVoiceLink.VoiceLink/TCVoiceLink.Application;TC/VoiceLink”. Remove if necessary.
- 4) Unregister the tcvlink.dll by running “regsvr32 /u c:\tcoss\system\tcvlink.dll”
- 5) From the directory “c:\tcoss\system\”, delete the files “tcvlink.dll” and “Error2BreakCode.txt”.
- 6) Remove the complete sub-directory “c:\topcall\shared\voicelib\tcvlink”.

10.9 Removing the TC/VoiceLink2 Version 1.0 and 1.2

This chapter describes how to remove the components of the VL2 which got obsolete with version 1.2 of the VL2.

- 1.) Make sure neither TC/VLAgent nor TC/VLANotif are running.
- 2.) Delete the registry keys “..\TOPCALL\TCVLAgent” and “..\TOPCALL\TCVLNotif”
- 3.) Remove the values “TCVLAGENT” and “TCVLNOTIF” from the registry key “..\Topcall\Boot\Startup”
- 4.) From the directory “C:\tcoss\system\”, delete the files:
 - o Tcvlagent.exe
 - o Tcvlagent.dll
 - o Tcvlanotif.exe
 - o tcvlaresolver.dll
 - o TCVL2PollFile.dll

10.10 Fault Tolerant TC/VoiceLink 2.2 Installation

This tutorial describes how to perform a fault tolerant installation based on the VL2.2 module update.

As a first step, perform a fault tolerant installation of TC/VoiceXML from KCS 7.80.04 or higher. Detailed installation instructions can be found in the Voice xSP manual [\[10\]](#). (Simply substitute VoiceAccess with VoiceXML.)

For the following steps assume that the primary VS instance is called TCECP01 and the secondary VS instance is called TCECP02.

After TC/VoiceXML has been installed successfully, install the module update according to the included readme file for TCECP01. Because the included “.reg” files point to the registry key “..\\TCECP”, replace all occurrences of “TCECP” by “TCECP01” within these files. Additional information about upgrade scenarios can be found in the chapter [Upgrade or Downgrade](#). The configuration description of the various features can be in the chapter [Configuration - Features - TC/VoiceLink2](#).

After TCECP01 is up and running, do the same for TCECP02.

As the final step, check if the error 12 is configured correctly for both VS instances. The correct configuration would be:

- Within the err2break.xml file the error 12 must have a “state” and “laststate” of 200. See [Enhanced Message Termination](#)
- The timeout mapping for error 12 should be > 10 sec. See [VL2 - Timeouts](#)

Naturally, both VL2 should poll the same queue, so just check to be sure.

If both VS have been configured correctly according to the Voice xSP manual [\[10\]](#), everything should work now.

11. Security

11.1 Used IP Ports

Depending on the configured protocol, the TCE_Rem may use different ports. Per default it uses the windows RPC service (ncalrpc) which uses a dynamic port range. The following link describes how to change this: <http://support.microsoft.com/kb/908472>

If the TCE_Rem is configured to use TCP/IP (ncacn_ip_tcp) the port 135 will be used.

If using VoiceXML, it may be necessary to open the port 80(http), 443(https) and 21(ftp). In some special situations, other ports are required. This situation can occur if a URL does not use the standard ports.

12. Audio Formats

The VS or to be more specific TCRT does support different audio formats. It is possible to convert these formats.

TCRT Name	Real name	Resolution	Sample rate	Bit rate (Byte rate)
PCM8	PCM	8bit	8kHz	8kB/s
PCM16	PCM	16bit	8kHz	16kB/s
G711_MULAW	G711 µ-law	8bit	8kHz	8kB/s
G711_ALAW	G711 a-law	8bit	8kHz	8kB/s
GSM610	GSM 6.10	Unknown	8kHz	1.6kB/s
G729A	G729 A	Unknown	8kHz	1kB/s

Table 20 – Supported Audio Codec's

To specify which format should be used for recording the registry key "TCRT\StreamFormat" [REG_SZ] must contains the TCRT Name from Table 20.

Where PCM 8 is the most common audio format, it has an average quality. It should be used when there are problems with different operating systems where G711 is not supported.

PCM 16 is like PCM 8 a standard format that is supported on all platforms. PCM 16 has a better quality than PCM 8 but it produces 2 times more data. This can lead to problems with the available bandwidth and storage space for voice messages.

Both, G711 µ-law and G711 a-law, are **no** compression algorithm! The difference between a G711 codec and PCM 8 is that G711 uses a logarithmic algorithm instead of a linear one to provide a better sampling rate. Both formats have a better quality than PCM 8 whereas G711 a-law has a slightly better audio quality than G711 µ-law. G711 is supported by most popular platforms free of cost.

GSM 6.10 is a lossy compression algorithm with a good quality. It uses only 1/5th of the bandwidth PCM 8 would use. The license of GSM 6.10 is free of cost and is installed on most Windows systems.

G729 A has the best compression but has a poor quality. It uses only 1/8th of the bandwidth PCM 8 would use but of a worse quality. The license of G729 A is not free of cost and the codec must be installed additionally on every PC which should perform playback of a recorded G729 A file.

So, what codec should be used?

1. The codec must be supported by every platform that is supposed to play a voice message.
2. Although conversion is not a big deal between the audio formats, it should be avoided. Depending on the used telephony system, different formats are provided. The common format for ISDN is G711 µ-law.
3. Choose the best quality. Users may not complain that the Voice server has to convert the voice message but they may complain that the quality is poor.
4. If the bandwidth is a problem, an algorithm with compression should be used.

In the latest releases, the default format is "G711 µ-law". This is because "G711 µ-law" is used by ISDN as audio format.

Note: Both G711 codices are supported by most Windows platforms.

12.1 GSM 610 Support for Cisco Native Integration

Note: The Cisco native integration is deprecated. This chapter is here mostly for historical reasons.

In order to save sound files in a GSM 6.10 - 8 kHz sound format, the following parameters have to be configured in the registry:

VS using Cisco:

“...\\TCECP\\TCRT\\StreamFormat” [REG_SZ] = “GSM610”

“...\\TCECP\\TCRT\\CNITAPIRecvBlockSize” [DWORD] = 0x3c0

TC/Player:

“HKLM\\Software\\Topcall\\Common\\TCRT\\StreamFormat” [REG_SZ] = “GSM610”

“HKLM\\Software\\Topcall\\Common\\TCRT\\DataBlockSize” [DWORD]= 0x780

For the VS using LS1 these settings are not applicable. GSM610 can be used for storing without changing above settings.

Background:

The GSM 610 format has a rather large BlockAlign size (65 compared to 1 or 2 of other formats). This means that the block size of the streamed data has to be a full multiple of that size. If the size does not match exactly, the rest of the block will only be silence leading to a hacking sound in the recorded or played file. Thus, the block size parameters have to be changed to the above values.

13. Supported TTS Engines

Although SAPI 4 and SAPI 5 are supported by the VS, it is recommended to use SAPI 5.

TTS Engine	Interfaces	Languages	Quality
Microsoft TTS	SAPI 4 + SAPI 5	English	Poor
Acapela Telecom	SAPI 4 + SAPI 5	Czech, Danish, Finnish, Faroe, Icelandic, Portuguese, Brazilian Portuguese, Russian, Turkish, South American Spanish	Very Good

Table 21 – Supported TTS Engines

Generally, every TTS engine which supports SAPI 4 or SAPI 5 can be used by the VS. But before using any 3rd party TTS engine with a productive VS extensive stress test should be performed. Some TTS engines might have problems under heavy load.

Note: The VL2 does not SAPI 4 anymore.

14. Telephone Systems

Currently, ISDN (LS1), H.323 and SIP (FoIPv3) as telephone system technologies are supported.

For details about the supported flavors and standard of ISDN, refer to [\[3\]](#).

Additional information about the supported H.323 and SIP telephone systems can be found in [\[23\]](#) and [\[24\]](#).

15. Performance

The following statistics are only parts of the performance tests documented in [\[11\]](#).

Of course a performance test can only reflect the behavior of a certain scenario at a given time. It is most unlikely that the tested scenario is exactly the same as the one of a customer. Maybe it is quite close to a user scenario but there are always some differing parts. These parts may influence the performance.

The scenario for the performance data of this chapter is a variation of the VoiceXSP scenario.

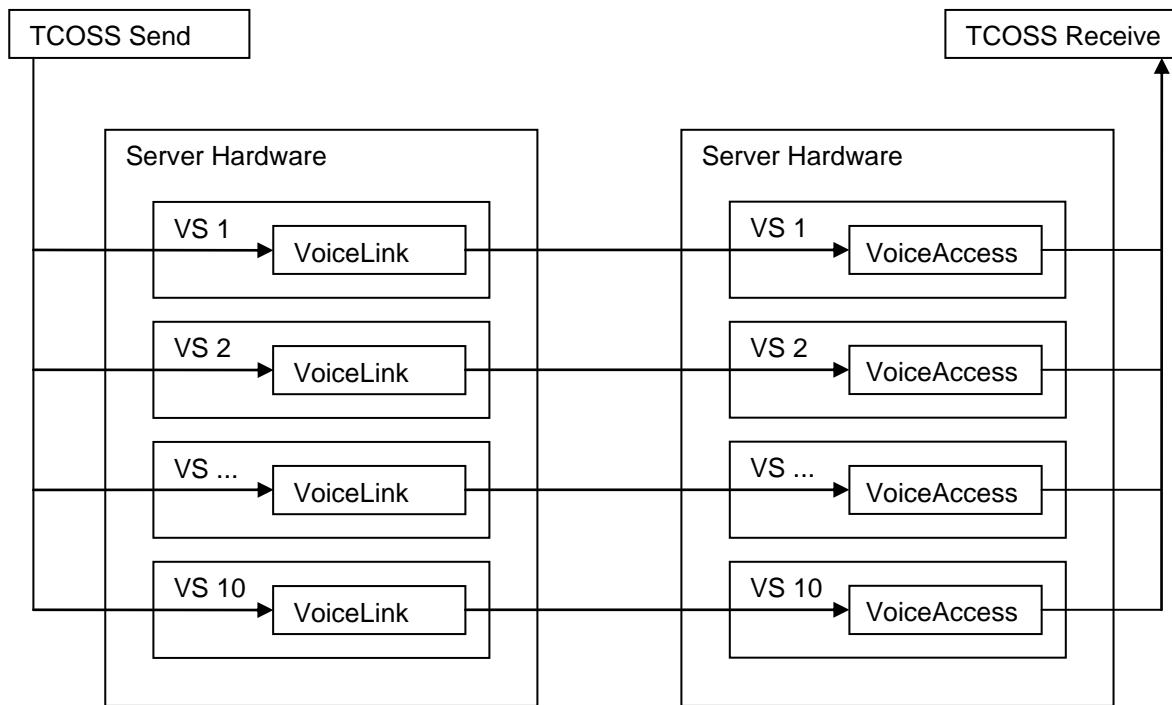


Figure 80 – Performance Scenario

10 VS instances on double processor server hardware run multiple VL instances. These VL2 poll the “Send” TCOSS. If a message is found, the VL2 will connect to a VA via LS1. On the receiving side, there are also 10 VS instances where each VS instance is related to a VS instance on the sender server. For more details, please read [\[11\]](#).

And now the statistics:

The servers are double processor 2,8GHz and 2GB RAM

The test description uses the following syntax: N * Object – options.

- N... Number of concurrent running object instances. (Where one VS instance runs a 1/10 of this number)
- Object... Either VL or VA
- Option... The following options are possible:
 - noprompts ... The VL was configured not to play any tts prompts (see [Disable TTS](#))
 - wav ... The VL message included a wav file
 - txt ... The VL message included text (130 words)
 - cached... The resource engine was configured to use its cache.

VA was only tested leaving voice messages without attendant. If this does not seem good enough please consider that normally VA is used by over 90% to leave messages.

The used TTS was MS-TTS with SAPI 5

The numbers given for CPU load and network load are averages over ~5 hours.

Test description	CPU Load [%]	Network load [%]	Comments
100*VL-noprompts.wav	3,5	8	
150*VL-noprompts.wav	12,6	35	
200*VL-noprompts.wav	17,7	48	
250*VL-noprompts.wav	22	59	First problems with the network occurred. Peaks in the network load lead to too short messages
300*VL-noprompts.wav	28	70	
100*VL-wav	15,8	20,8	With growing load the playback time starts to vary very much. This is caused by the playback of the prompts via TTS.
150*VL-wav	24,7	31	
200*VL-wav	34	41	
250*VL-wav	43	53	
300*VL-wav	52	59	
100*VL-txt	48,6	7,4	As shown, the CPU-load increases dramatically when only TTS is used for playback. As a result, errors will appear. (100*VL ~ 0,3% errors; 150*VL ~1,4% errors; 200*VL ~ 12,4% errors)
150*VL-txt	74	11	
200*VL-txt	99,3	14	
100*VL-txt,cached	13,7	7,4	Caching helps to decrease the CPU load even below the level if only .wav – files are played. This is because also the necessary audio-codec conversion is cached. Although everything looks fine, caching introduced new problems (see Restrictions - TTS Conversion Caching)
150*VL-txt,cached	11,2	11	
200*VL-txt,cached	13	15	
250*VL-txt,cached	18	18	
300*VL-txt,cached	20,4	22	
100*VA	5,4	5	
150*VA	25	22	
200*VA	36	29	
250*VA	49	35	
300*VA	64	40	

Table 22 – Performance Data

16. Known Issues

16.1 Restrictions

16.1.1 General

- TCECP monitors all connections and will terminate them if no activity is detected after a specified time. Unfortunately, the time during playback or recording does not count as activity. This may lead to the situation that a call is terminated although the user is still listening or recording. The timeout when TCECP terminates a connection can be configured in msec via the registry key "General\WorkerTimeout" [REG_DWORD].

16.1.2 TCRT

- You can only use the TTS Speaker of one interface. (SAPI 4 or SAPI 5)
- It is not possible for TC/ECP to check if there are enough valid licenses for the TTS engines used. Neither the Microsoft SAPI 4, nor the SAPI 5 Interface provides a method for such a check. So if start-up fails, this can be due to missing licenses, however there is no guarantee that if start-up works, the licenses are ok as well. Therefore, you should check the test files that are created at start-up ("ResLXStartup*.wav") after installation or after configuration changes.
- It is recommended not to start TTSCConf.exe while TC/ECP is running. Some TTS engines (e.g. Loquendo) will not be shown if TC/ECP is active.

16.1.3 TTS

- You can only use the speakers of one interface.
- It is not possible for TCECP to check if there are enough valid licenses for the TTS engines used. Neither the Microsoft SAPI 4 nor the SAPI 5 Interface provides a method for such a check. So if start-up fails, this can be due to missing licenses, however there is no guarantee that if start-up works, the licenses are ok as well. Therefore, you should check the test files that are created at start-up ("ResLXStartup*.wav") after installation or after configuration changes.
- It is recommended not to start TTSCConf.exe while the VS is running. Some TTS engines (e.g. Loquendo) will not be shown if the VS is active.
- If using SAPI 5, there are no language specific fall-back TTS speakers. If some TTS speaker is not configured correctly, simply the first available is used. The registry keys ...\\TCRT\\TTSSpeaker01, TTSSpeaker02... are not used.
- Some TTS engines will not speak the text if it contains "unknown" characters. Such an unknown character might be e.g.: "€". In such a case, TTS will produce only silence. It is possible to test if a TTS does not speak a certain character. To do this, run the "Speech" application on the Control Panel. Then, click on the tab "Text To Speech" and select the TTS engine to test from the drop down list. Afterwards, past the text into the input field and press "Preview Voice". If the complete sentence is played, the TTS engine does not have a problem. Note that the TTS will only play a limited number of characters.
- Depending on the credentials of the used accounts, different process cannot access the same TTS speakers. In such a situation the process want start and the error 80070005 appears in the trace file.

16.1.4 TC/VoiceAccess

- If a Kofax Communication Server user deletes a message, it will only be terminated positive. The message will not be deleted.
- USN is not supported by Cisco CallManager
- Loop detection is not supported with CISCO CallManager
- "Internal" call transfer is not supported with CISCO CallManager
- DistList and DBN only supports recipients and DistLists from the system address book
- Read notifications are supported for MS-Exchange only (and partly by Lotus Notes):
In the Send-Options menu (Standard key assignment 70), you can request a read notification for a voice mail that will inform the sender when the mail-recipient actually opened (listened to) the message. This functionality currently is fully supported for MS-Exchange only. For Lotus Notes following limitations apply:
If a read notification is requested, there are three possible scenarios:
 - The message is opened within the Notes Client. No read notification is sent.
 - The message is listened to via phone. The originator receives a read notification.
 - After the message was opened within the Notes Client, the message will be set to unread again. When the message is now listened to via phone, a read notification will be sent.
- Urgent flags are currently supported only if originator and recipient belong to the same mail system:
The send option Urgent only works if originator and recipient belong to the same mail system (User belongs to field in the user profile).
- Mark/Unmark messages as read is not supported with the TCOSS-mail system:
The message option "Mark/Unmark message as read" is only available if the mail system is MS-Exchange or Lotus Notes.
- The error #9825 was only corrected for "CallSender". For "CallOperator" or "DialByName", the timeout has to be set sufficient long.
- The functionality "Navigation in Emails" currently does not work.
- Infovox 330 has some restrictions with special characters or character combinations. See Error #7425
- Standard VP are not supported with the TC/MC. If the TC/MC opens a user with a standard VP, it will be displayed as an empty VP. If the TC/MC saves a user with a standard VP, the saved VP will be empty.
- Account locking has no impact on calls from a full-auto-login address. These calls can still access the mailbox even if it is locked.

16.1.4.1 TC/PreProcess

- Replacements and conversions can by design only be applied to single tokens. Whitespace and typographic separators like dots or commas separate tokens.
- Note that no spaces, tabs, dots, commas or other list separators can be used in entries in tcprep.mpf, tcprep.rul and tcprep.dct.
- URLs and email addresses are handled as single tokens, although they can include separators like dots.

16.1.5 TC/VoiceLink2

- If the VL2 is connected remotely to a voice server, the TTS-engine has to be installed local.
- For TTS, always the local resources are used.
- The VL2 does not use any pre-recorded prompts because of this and the poor quality of the MS-TTS engine, the default may not sound very satisfying.

- Some telephones like the Siemens Optiset E Basic, will not provide DTMFs when they are called. So VL2 will not work with these telephones.
- The file name of a template must not contain any non ASCII character “€”. “Umlaute” are allowed.
- When sending with the old send option syntax, the message must be of type „text”.

16.1.6 TC/SPI-LN

- If two users (one of them is the SPI) open a database and modify the read/unread-flags, the user who will close the database at last, will change the read/unread-flags, the other changes will be ignored!
- When setting the filter “State”, this filter will be taken for all documents. It is not possible to define this filter in a sub-filter. For example, this filter is NOT possible: “DocClass = 16 AND State = 800 OR (DocClass = 16 AND State = 700). Only one “State” should occur in the filter expression.
- TC/SPI-LN must not run on the same as TC/LINK-LN. TC/LINK-LN kills Notes processes, if an error occurs which will also kill the Notes API and so the TC/SPI-LN will not have a valid Notes session anymore!
- If a user has a new mailbox on the Notes server, it is necessary to install a client. Open the mailbox of the user and close it. This is necessary to get the correct unread/read list for the SPI user. Otherwise, the SPI-user is not able to get the correct READ/UNREAD-flag status. (Or press Shift-F9 to refresh the view – also the read/unread flags!)
- If the MetaMail functionality is used, TC/Link-LN 2.04.04 or higher is required.
- If the mailboxes have a newer ODS (on disc structure) version then the Notes Client supports, this can lead to unexpected behavior (e.g.: read/unread marks). This can only be solved by updating the Client on the Voice Server.
- In Lotus Domino cluster environment: when a user plays back a message, the unread mark for this user will be removed from this message. But this information will only be propagated correctly among the mail-db-replicas in the cluster, if the database property “Replicate unread marks” (see on the last tab on the Database Properties dialog box) is set at least to “Clustered servers only”. (The unread mark propagation is important, because otherwise, in case of a failover, the alternate server and db-replica will be used and so already heard messages could be played back as “unread” message.)

16.1.7 TC/SPI-MX

- Only 1 Exchange organization can be accessed. If access to another organization is needed, please add a second voice server.
- TCSPI Exchange can only access the folders “Inbox” and “Sent Items” and the Global Address Book. Access to any other folders (e.g. public folders, Contacts etc.) is NOT SUPPORTED.
- If run in server mode, all configuration settings stored on the mailbox owner’s client workstation will be ignored.
- Exchange clusters are supported but it is not allowed to run TCSPI Exchange itself on a server that is part of a cluster.
- When sending a message via TC/VoiceMail, you can order a read notification for all recipients. Read notifications will only be created for Exchange recipients. For all other recipients, the originator will find a delivery (or non-delivery) notification in the inbox. The (non)delivery notification is ignored by TC/VoiceMail.

- If a message was deleted before being read, Exchange creates a non-read notification. The originator then will find the non-read notification in his inbox. The non-read notification will be ignored by TC\VoiceMail.
- TCSPI Exchange uses a privileged account to log on to any Exchange mailbox. Whenever a mailbox is opened, the Exchange Information store writes an application event log entry # 1016 ("User xxx logged on to user yyyy mailbox, and is not the primary Windows account on this mailbox"). According to Microsoft, this behavior is by design and cannot be disabled.

16.1.8 TTS Conversion Caching

The caching feature of the tce_res helps to minimize the CPU load dramatically by caching the output of a TTS or audio conversion. The drawback is that the tce_res will not notice if the TTS output is correct. If the first TTS conversion fails because of the currently heavy load of the VS, the cached result will be wrong and all subsequent calls to the cache will return a bad audio file.

This leads to the paradox situation where TTS caching should not be used in a high load/volume environment.

There is no solution for this problem but only a manual workaround for the VL. Before sending a message to x*1000 recipients, send the message only to one. This will allow the VS to cache the result without being effected by TTS conversion problems. When now sending the rest, the correct results can be taken from the cache.

16.2 Troubleshooting

16.2.1 Corrupt TTS output with ELAN TTS 4.6

On a test system, the trace showed errors regarding the failure of setting pitch and speed of the Elan Speechcube 4.6 TTS engine. The first TTS conversions sound ok but then, the wave files consist of TTS output that seems to be due to the different volume, pitch and speed settings of the Elan TTS speaker.

Solution:

If the registry keys for pitch and speed are set to maximum, the pitch and speed (rate) setting will be skipped and the Elan TTS works fine:

Set the key "TCRT\TTS\GUID\Pitch" [DWORD] to "0xFFFFFFFF"

And the key "TCRT\TTS\GUID\Rate" [DWORD] to "0xFFFFFFFF"

16.2.2 TC/SPI-LN

- **Cannot login:** Check Kofax Communication Server user again (belongs to, voice address). Also check if the registry path "...\\SPI\\Notes" exists, maybe the DLL was not loaded by TCECP. Possible reason: TCSPILN.DLL was not registered. Try to register the SPILN.DLL – there you will get further informations about the error (maybe Notes Path not set in the environment!)
- **Login ok, but "Your command could not be processed":** The client on the VoiceServer (with the ID-file) does not have manager rights! Check this in the Access Control List (ACL) of the user's mailbox.

- **Login ok, but no voicemails:** belongs to is wrong (set it to NOTES) or the registry key “...\\SPI\\Notes\\VoiceExtensions” does not contain the extension (e.g. “.wav”) to filter the voicemails.
- New voicemails (leaving messages via telephone) is not working: In-Events to Notes in the Kofax Communication Server user are not configured.

16.2.3 Acapela TTS

Acapela TTS plays silence:

There have been reported problems with the Acapela TTS version 6.0.1.0, where the TTS produces zero length wave files. On the telephone this behavior may result in an immediate disconnect.

There exists an update for Acapela to version 6.0.1.1 which is attached to the hint #13561.

Acapela TTS Uses Very Long Before Playback Starts:

Acapela does use a lot of resources for TTS conversion and therefore it might happen that it takes some time that the called person does not here anything for quite a while before the playback starts. This time can be reduced to almost 0 if the feature [Text2Wave Pre-Conversion](#) is used.

16.2.4 The VL2 Does Not Play Umlauts When Using Acapela TTS

This is actually a bug in TC/VoiceXML (tcvxitiprompt.dll 1.00.05), but as a workaround it is possible to activate the feature [Text2Wave Pre-Conversion](#).

16.2.5 TC/VoiceLink2

- Messages are set to “inactive problems” or “canceled” before the call was placed, when running in a fault tolerant installation:
Check that the “err” 12 in the “err2break.xml” file has a “state” and “laststate” of 200.
- **In a fault tolerant installation all messages are permanently opened and closed although the active VL2 has no free channels:** The most likely reason is that the VL2 which should be on stand by tries to send messages and because he does not succeed he immediately tries the next message. To prevent this, set the timeout for error 12 to something high like 30000 (30s). This means that the VL2 in stand by will retry to send the next message in 30 seconds.
- **The VL2 does not play umlauts when using Acapela TTS:** This is actually a bug in TC/VoiceXML (tcvxitiprompt.dll 1.00.05), but as a workaround it is possible to activate the feature [Text2Wave Pre-Conversion](#).

16.2.6 Fault Tolerant Voice

Please refer to the troubleshooting chapter of the Voice xSP manual [\[10\]](#).

16.2.7 Bad TTS Audio Quality

In the scenario where an Acapela TTS engine is used in combination with the text2wav pre-conversion feature of the VL2, the audio quality might not be as good as required. To solve this issue, change the registry key “..\\TCECP\\FileAudioFormat” [DWORD] to “1”.

17. Hints

17.1 TC/VoiceAccess

- There may be a situation where VA seams to start, but it does not pick up the call. This is maybe the result of a change in the DCOM rights setting. For details please see the hint/error #9777.
- The telephony system must provide the correct and dial able number of the caller; otherwise the call sender feature may not work properly.
- It may occur that the system detects an email as voicemail or a voicemail as email. This has no influence on the functionality
- For USN, all channels must support fax AND Voice!
- Only a reply to an email or to a Voicemail box works. There is no guarantee that replies to fax machines or SMS will work.
- The reply of voice mail (from non-Kofax Communication Server users) should be rejected. Otherwise, the reply will lead to unprocessed send order in Kofax Communication Server. The problem can be solved by using the auto reject feature (enh. 4848, see Kofax Communication Server release documentation for further details). You have to use KCS 7.35.00 or higher. Create a user "V" with flag "reject incoming" set. This user must not have in-events. The dependences created by c:\tcoss\system\vmaildep.bat were changed in KCS 7.36.00 to create this "V" user. Since vmaildep.bat does not change existing users or service, it can even be used for upgrade of voice installations. The User "V" is a valid originator but sending to this user will lead to a non-delivery notification.
- If a customized prompt should be played (e.g.: song1.wav), special care should be taken when updating the installation. It is recommended that the customized prompt is saved in a separate directory (e.g.: C:\topcall\shared\customized\01) and not in the VoiceLib (e.g.: C:\topcall\shared\voicelib\tcvmail\01) because this directory will be overwritten! However, there are 2 possibilities to use customized prompt.
 - 1 (recommended): Overwrite the original shipped prompt with the customized prompt after the update.
 - 2: Add the path of the directory to the registry key "TCRT\ResDefaultPath), see also [Languages](#)

17.1.1 TC/PreProcess

- The time in milliseconds the processing takes will be traced after the processing if the **info trace** (level 0x2) is activated. Note that failed MetaMail searches (especially when accessing an archive server) can need most of the time.

17.2 TC/VoiceLink2

- If using custom break codes, be aware that if the retry count reaches 1, VL2 will automatically terminate the message and set the error code.

- Messages that are not automatically terminated have to be terminated manually. Otherwise, there can be problems with too many open messages.

17.3 TCRT

- If the TTS playback stops before the whole text has been played, check if the value of the registry key “TCRT\TTSBufferSize” is big enough. Approximately 16kByte data are required per second speech.

17.4 TTS

- If it is not possible to configure the same speaker for two applications at the same time, check if both applications use the same user to start (E.g.: “Local System Account”). In most cases, this situation is indicated by an “access denied” error (error code 80070005) in the trace file.

18. Upgrade or Downgrade

18.1 From VL2 1.0/1.1 to VL2 1.2 (aka VL2.2)

When migrating from previous version of the VL2 to the version 1.2 (also known as VoiceLink 2.2) you must be aware that several features are not supported any more.

Features which are not supported any more:

- “VoiceLink2 Configuration Client” (the whole configuration is now done again via the registry)
- File poller
- (In)activity scheduler
- Voice address verification
- Broadcast jobs
- Broadcast job reports
- Distribution lists

Basically, it can be said that the whole native broadcasting functionality is no longer supported. If broadcasting is required, the standard TC/Broadcast product must be used.

It is assumed that the TSCI Poller was used. Furthermore, it is recommended to stop the VS, the TC/VLAgent and TC/VLANotif before starting the migration.

As a first step, it is necessary to update the KCS to version 7.80.04 or higher. During the setup, choose to upgrade the “TC/VoiceXML”, but deselect the “VoiceLink Business Connector”.

Now install the module updates for the VL2 as described in the readme file.

The VL2 version 1.2 uses the same default values as the previous versions of the VL2. I.e., it should be enough to only migrate the changed values to the registry.

- 1.) Open the “VoiceLink2 Configuration Client”.
- 2.) If the configured poll cycle differs from the standard poll cycle of 10 seconds, it can be copied directly to the registry key “...\\TCECP\\TCVLink2\\PollCycle”. See [Poll Cycle](#)
- 3.) The value of the “Name of the remote voice server” can be copied directly to the registry key “...\\TCECP\\TCVLink2\\VoiceServerName”. See [Remote Voice Server](#)
- 4.) A different “IVR plug-in alias” must be configured via the registry key “...\\TCECP\\TCVLink2\\VoiceXMLAlias”. See [Individual IVR Plug-in](#)
- 5.) If the value of “Mailbox detection” is changed it can be copied to the registry key “...\\TCECP\\TCVLink2\\Mailboxdetection”. See [VL2 - Answering Machine Detection](#)
- 6.) If timeouts have been changed they need to be merged into the registry key “...\\TCECP\\TCVLink2\\Timeouts”. See [Timeouts](#). Please note that the “timeout code” is referred as “error code in this chapter.”
- 7.) Check whether TCSI Poller has the same TCOSS configured as it is configured in registry beneath the sub-key “...\\TCECP\\Topcall\\”. If this is not the case, you need to configure the VL2 to use an alternative TCOSS. See [TSCI Poller – TCOSS configuration](#)

- 8.) Most likely you will also need to change the queue which the VL2 polls or the service which is used to address this queue. You can change the name of the queue with the registry key “...TCECP\TCVL2PollTcoss\Queue” See [VL2 - TCOSS Queue](#)
- 9.) If the “Log entry name” of the TCSI Poller is neither empty nor “default” this has to be migrated too. See [Pollers](#) for a detailed description.
- 10.) The value of “Dialout prefix” and, if visible, “Internal prefix” can copied directly to “...TCECP\TCVL2PollTCOSS\DiaOutPrefix” and “...TCECP\TCVL2PollTCOSS\InternalPrefix”. See [Telephone number prefixes](#)
- 11.) If a different PC code page than the standard PC code page was used this has to be migrated too. See [TCSI Codepage Conversion](#) for details.
- 12.) A different location of the “Error codes file” must be configured with the registry key “...TCECP\TCVL2PollTcoss>Error2BreakFile”. See [Enhanced Message Termination](#)
- 13.) Any configured “Recipient fields” must be migrated to the registry key “...TCECP\TCVL2PollTcoss\RecipientFields”. See [Recipient Properties](#)
- 14.) If a logger was configured, the logger has to be re-enabled and configured. See [Logging & Billing](#)

That concludes the configuration via “VoiceLink2 Configuration Client”.

To migrate the configuration of the TC/VLA Transformer to the VL2 it is enough to copy the whole sub key “...\\TCVLAgent\\VLATransform” to “...\\TCECP\\VLATransform”. Now you need to do the following changes.

- 1.) If there had been configuration within “...\\TCVLAgent\\Server1\\” which was intended to overwrite the configuration of “...\\TCVLAgent\\VLATransform”, this configuration has to be merged into “...\\TCECP\\VLATransform”.
- 2.) Disable the text2wav conversion by setting the registry key “...\\TCECP\\VLATransform\\Convert2Wave” to “0”.
- 3.) Make sure that the value of “...\\TCECP\\VLATransform\\BaseUrl” is empty.

As one of the last steps, check if all directories are configured correctly. See [Directories Used](#)

Finally, you might want to uninstall the “VoiceLink2 Configuration Client” and do not forget to disable the TC/VLA Agent and TC/VLANotif. (To do this, remove the values “TCVLAGENT” and “TCVLANOTIF” from the registry key “HKLM\\Software\\Topcall\\Boot\\Startup”.)

18.2 From VL to VL2

The new architecture of the VL2 does not allow a full 1:1 mapping of the configuration from VL to the VL2. This chapter only describes a basic migration. If features like “NoPrompts” or “SendEmptyMsgs” had been used, it is necessary to have an in-depth look at the new features and possibilities to emulate this behavior.

It is recommended to perform the migration while the VS is stopped!

Although the VL2 should run without any configuration changes, it might be necessary to adapt certain options.

- 1.) Install TC/VoiceXML from KCS to 7.80.04 or higher. Do not choose to install the “VoiceLink Business Connector” too.

- 2.) Install the VL2 module updates as described in the readme file.
- 3.) Make sure that the VL2 as TCECP plug-in is configured correctly. See [Plug-In Configuration](#)
- 4.) Delete the registry key "...\\TCECP\\TCRT\\ResCacheProtocols". This will enable more different protocols to be cached.
- 5.) The VL allowed the configuration of another TCOSS server than the one configured in the "...\\TCECP\\TOPCALL" registry sub key. This alternative server was configured via the registry keys "...\\TCECP\\TCVLink\\Server_for_VLink", "...\\TCECP\\TCVLink\\ServerPath_for_VLink", "...\\TCECP\\TCVLink\\VLink_User" and "...\\TCECP\\TCVLink\\VLink_Pwd". The VL2 offers the same possibility. See [TSCI Poller – TCOSS configuration](#)
- 6.) If the queue for the VL has been changed to something different than "VL", this has to be changed for the VL2 too. To do this, copy the value "...\\TCECP\\VLink_Queue" to "...\\TCECP\\TCVLink2\\Queue" See [VL2 - TCOSS Queue](#)
- 7.) To adapt the default behavior where no send options are specified, the VL2 uses two registry keys. The key "...\\TCECP\\VLA\\Transform\\Menu1" [DOWRD] which equals the old key "...\\TCECP\\TCVLink\\askatfirst". The key "...\\TCECP\\VLA\\Transform\\Menu2" [DOWRD] which equals the old key "...\\TCECP\\TCVLink\\askaccept".
- 8.) The value of key "...\\TCECP\\TCVLink\\PollCycle" can be safely copied to "...\\TCECP\\TCVLink2\\PollCycle". See [Poll Cycle](#)
- 9.) In case the VL used a remote VS the VL2 can be configured to do the same. Copy the value of the key "...\\TCECP\\TCVLink\\VoiceServerName" to the key "...\\TCECP\\TCVLink2\\VoiceServerName". See [Remote Voice Server](#).
- 10.) If the VL used different codepages than the standard ones, please check [VL2 - TCSI Codepage Conversion](#)
- 11.) If you have customized the prompts for the VL, you have to edit the file "OldVoiceLink.vxml" which is located in "C:\\tcoss\\system\\VLA\\templates". Note: This has to be done with an editor supporting UTF-8.
- 12.) If you customized the "error2break.csv" file, it is also necessary to adapt the new "err2break.xml" file (see [Enhanced Message Termination](#))
- 13.) Start the VS.

For information about how to uninstall the VL, please refer to [Removing the Old TC/VoiceLink](#).

18.3 From SAPI 4 to SAPI 5

The following steps are necessary to update the VS from Microsoft SAPI 4 interface speakers to Microsoft SAPI 5 interface speakers.

- Stop TC/ECP
- Install TTS speaker (e.g. from Loquendo, Elan, Bable Infovox,...); if asked for the interface, you have to choose Microsoft SAPI 5; activate the necessary licenses for the TTS speakers
- If the installed KCS is older than 7.59.00, install a newer release.
- Start TTSConf; Switch to SAPI 5; for each desired TTS speaker, set the number of instances and set the speaker to active; save the settings by clicking OK

- Start TC/ECP; for each instance of each configured TTS speaker, an example conversion will be done and written to "C:\TCOSS\TRACE" with the name "ResLXStartup*.wav"

As you have changed your TTS speakers, you have to update your TTS speaker specific configuration. This means especially the **Speech profiles** of the Kofax Communication Server **TCVMAIL user** (the user configured in the registry with "...\\TCECP\\Topcall\\User") have to be changed. Each user with special TTS settings also has to update his/her Speech profiles.

- Use the TC/MC to update the Speech profiles of the TCVMAIL user. Note: The speakers with the numbers 5 – 8 can be used by all users by choosing 91 – 94 on the telephone. For using VL, TC/ECP has to be restarted after these changes (the TCVMAIL user settings are only read at start-up).
- For VA each Kofax Communication Server user with individual TTS settings has to update his/her personal Speech profiles (if used). These are the individual speakers that can be chosen by typing 95 – 98 while listening to the Emails on the telephone.

19. Documentation Changes

This chapter lists big changes in the documentation. Big changes are restructuring of a whole section or the removal of several chapters. It's no big deal if a feature is added, so it won't be listed here.

19.1 7.80.00 Edition 5 -> 7.80.04

Because of the dramatic changes in the architecture of the VL2, following chapters have been removed, or massively changed:

Chapter	Changes
4.14 Voice Link Business Connector	Renamed, restructured and content changes.
8.4.2 Installation – Setup – VLBC	Removed except for a short notice.
9.3.6.1 Introduction	Removed
9.3.6.2 Pollers	Restructured
9.3.6.14 (In)activity Scheduler	Removed
9.3.7 TC/VLAgent	Removed. The sub chapters have been merged into the TC/VoiceLink2 chapter 9.3.6.
9.3.7.1 Introduction	Removed
9.3.7.2 TC/VoiceLink2 Queue	Moved to chapter 9.3.6.4 TCOSS Queue
9.3.7.3 Voice Address Verification	Removed
9.3.7.4 Directories Used	Moved to chapter 9.3.6.17 Directories Used
9.3.7.5 Base URL	Moved to chapter 9.3.6.18 Base URL
9.3.7.6 Old VoiceLink Compatibility	Moved to chapter 9.3.6.19 Old VoiceLink Compatibility
9.3.7.7 Default Template	Moved to chapter 9.3.6.20 Default Template
9.3.7.8 Addressing to TC users	Removed
9.3.7.9 Message Properties	Moved to chapter 9.3.6.21 Message Properties
9.3.7.10 Keep Remainings	Moved to chapter 9.3.6.22 Keep Remainings
9.3.7.11 Default Job Type	Removed
9.3.7.12 Placeholder Identification	Moved to chapter 9.3.6.23 Placeholder Identification
9.3.7.13 Customized Send Options	Moved to chapter 9.3.6.24 Customized Send Options
9.3.7.14 Offline Text2Wav Conversion	Removed
9.3.7.15 Number Locking	Moved to chapter 9.3.6.25 Number Locking
9.3.7.16 Enhanced Job Termination Settings	Removed
9.3.7.17 Job-End And Job-Start Notifications and Reports	Removed
9.3.7.18 Alternative Notif User	Removed
9.3.7.19 Resolving Distribution Lists	Removed
9.3.7.20 Tracing	Merged with chapter 9.3.6.16 Tracing
19.2 From VL2 1.0 to VL2 1.1	Replaced by "From VL2 1.0/1.1 to VL2 1.2 (aka VL2.2)"

Table 23 – Documentation Changes 7.80.00ed5 – 7.80.04

20. Third Party Licenses

Following third party components and their licenses are used:

Component	License	Available at	Used by
Apache XerxesC	Apache License 2.0	http://www.apache.org/licenses/LICENSE-2.0	TC/VoiceXML
Expat	MIT License	http://www.opensource.org/licenses/mit-license.php	TCRT
OpenVXI	ScanSoft Public License 1.2	http://www.speech.cs.cmu.edu/openvxi/sspl.html	TC/VoiceXML
OpenH323	Mozilla Public License 1.0	http://www.opensource.org/licenses/mozilla1.0.php	VoIPv3
LibWWW	W3C Software License	http://www.opensource.org/licenses/W3C.php	TCRT
OpenSSL	OpenSSL License	http://www.openssl.org/source/license.html	TC/VoiceXML
Boost	Boost Software License 1.0	http://www.boost.org/LICENSE_1_0.txt	TC/VoiceXML
SpiderMonkey	Mozilla Public License 1.1	http://opensource.org/licenses.mozilla1.1.php	TC/VoiceXML

Table 24 – Third Party Licenses

If the supplied link to a license is broken, a copy of the license can be requested at Kofax.

21. Referenced Documentation

- [1] Voice_Programmers_Reference_10000.doc (This manual has not been published yet!)
- [2] VoiceLink_10203_User.doc
- [3] ls1_76500_man.doc
- [4] ~~tce_h323_10507_tech.doc~~—obsolete manual, see [23] and [24]
- [5] TCOSS_75906_sys.doc
- [6] TCOSS_75906_tam.doc
- [7] TCOSS_76301_isdn.doc
- [8] TCVAccess_30900_User.doc
- [9] TCDISTMF_10128_IMPL.doc
- [10] VoiceXSP_78000_tech.doch
- [11] DMS_LineServer and Performance testing.doc
- [12] MetaMail_ConfigManual_10100.doc
- [13] TCAt2Url_10301_man.doc
- [14] Model3xx_3002_man.pdf
- [15] TA11_0D.pdf
- [16] tcweb_40801_man_install.doc
- [17] tccp_51301_admin.doc
- [18] TCUserTool_10100_man.doc
- [19] tcweb_41000_man_user.doc
- [20] tcbroadcast_11000.doc
- [21] http://www.boost.org/libs/regex/doc/syntax_perl.html
- [22] tcoss_78000_cnf.doc
- [23] foip_man_xyzz.doc
- [24] IP_Integration_Guide_xyzz.doc

22. Terminology

22.1 Abbreviations

AS1	Application Server 1 this is double processor server hardware 2,8GHz and 2GB RAM
CCM	Cisco Call Manager
DBN	Dial by Name
DistList	Distribution List
DT	Day Type
E2B	Error 2 Break
EMT	Enhanced Message Termination
GG	Greeting Groups
GS	Greeting Scheduler
GUID	Global Unique IDentifier
GWIZ	Greeting Wizard
IVR	Interactive Voice Response. Is the more common word for TUI.
LRS	Language ReSource (File)
LS1	TC LineServer One
LSR	Local Session Redirection
msec	Milliseconds (1/1000 second)
N/A	Not Accessible or Not Available
SAPI	Speech Application Programming Interface
SPI	Service Provider Interface
SSML	Speech Synthesis Markup Language
SysUser	System User
TC/LP	Former KCS Link Package, now a part of Kofax Communication Server
TC/MC	TC Management Console
TC/SP	Former KCS Server Package, now a part of Kofax Communication Server
TCECP	TC Enterprise Communication Platform
TCfW	TCfW Communication Server Client
TCRT	TC Real Time
TCSI	TC Server Interface
TF	Time Frame
TFC	KCS Foundation Classes
TRS	Transparent Routing Script
TTS	Text to Speech
TUI	Telephony User Interface
USN	Unique Single Number
VA	TC/VoiceAccess
VL	TC/VoiceLink
VL2	TC/VoiceLink 2
VP	Voice Profile
VS	Voice Server

Table 25 – Abbreviations

22.2 Glossary

Call Transfer	Although Call Transfer is a feature, it also describes the process whenever VA establishes a second connection to another user or mailbox
Caller	A caller is a person who wants to leave a message on a mailbox
Codec	Although codec is an abbreviation for "CCode/DECode", it means a program retrieving data and transforming it. E.g.: A specific audio codec may retrieve PCM 16 audio data and transforms it into PCM 8 audio data.
Day Type [DT]	A set of time frames describing the behavior of the VL2 during a day.
Directory Synchronization	Process where a TC/Link synchronizes the users between TCOSS and another mail system. In most cases this means that the TC/Link adds, removes or modifies shadow users in TCOSS in a way that they represent the state of the other mail system
Greeting Groups [GG]	Is a pointer from the user's voice profile to a voice profile of the SysUser
Greeting Scheduler [GS]	Is a special voice profile that allows a user to configure a week day and a time when to play a specific greeting
Greeting Wizard [GWIZ]	A wizard that is played the first time a user logs in his mailbox.
Interactive Voice Response [IVR]	Describes a system which reacts on the input of the recipient. This term is closely related to the TUI
Local Session Redirection [LSR]	A way a plug-in can tell TCECP that a call should now be processed by a different plug-in
Lossy Compression	A way to compress audio data very well but with the disadvantage to lose some information. Because of this, lossy compressions have worse quality than the original.
MetaMail	A technology to forward links to large messages stored on TCOSS to third party mail systems and to open the original messages by resolving the link. MetaMail resolving is supported by TC/VoiceAccess when TC/PreProcess is enabled.
Proxy Address	An inactive address in the user profile of a shadow user. The proxy address is the address of the user in the mail system the user belongs to.
Read Section	After a user has listened to all the new messages in the mailbox, he will hear: "You have XXX already heard messages". All these already heard messages are in the read section.
Service Provider Interface [SPI]	Software allowing TC/VoiceAccess to talk with other mail systems than TCOSS. The SPIs are so called plug-ins for TFC and cannot be accessed directly. The term SPI is used to refer to the specific plug-in. But it is also commonly used to refer to a mail system that is supported by such a plug-in.
Shadow User	A shadow user describes the user profile of a user in TCOSS who belongs to a different mail system. Shadow users represent users of different mail systems in TCOSS.
Speech Application Programming Interface [SAPI]	Microsoft standard to communicate with various TTS engines
System User [SysUser]	The user, TC/VoiceAccess uses, to communicate with the TCOSS – Server
TC/Player	Client software that is capable to play and record audio files via soundcard or telephone.
TC/VoiceAccess [VA]	The main Voice Application.
Telephony User Interface [TUI]	Is the audio counterpart of a GUI (graphical user interface)
Text to Speech [TTS]	A technique converting text into sound

Time Frame [TF]	A period of time which has a starting and ending time. During this period the VL2 is merely not active or not at all.
TC Enterprise Communication Platform [TCECP]	The core component of the VS. Actually it is the process managing the various plug-ins.
TCfW	Client mail and administration program. Used to administer users and Kofax Communication Server.
TC Foundation Classes [TFC]	An API that can be used to access different mail systems using a uniform and consistent abstraction layer. If TFC is used native, Kofax Communication Server is the only possible mail system. To connect to other mail systems, the SPIs are needed.
TC Language ID	Is a 2-digit-number representing a language. E.g.: "01" is English, "02" is German or "03" is French.
TC Management Console	Client program allowing the user to administer his/her own settings. The Management Console is part of the KCS Client Applications.
TC Real Time [TCRT]	Is the hardware abstraction layer for the VS.
Transparent Routing Script [TRS]	Script which decides which plug-in should handle an incoming call.
TTSCConf	A program distributed with KCS allowing configuration of the TTS engines on the Voice server.
Unread Section	If a user accesses his mailbox, he will hear "You have XXX new messages." These new messages are in the unread section.
User	A person who uses or has something to do with voice access. This can be directly or indirectly. In most cases, it will refer to a person wanting to access their mailbox and listens to the messages in it.
Voice Profile	A voice profile defines which greeting is played for internal or external callers and if the caller can leave a message.

Table 26 – Glossary

23. Configuration Parameter Overview

ATTENTION: The following configuration parameters are only mentioned here because of documentation reasons. If a configuration parameter is not mentioned in the documentation above, only change values if asked by the support!

23.1 General

Registry Key	Type	Default	Comment
Commandline	REG_SZ	TCECP TCECP	The command that is executed to start an instance of TC/ECP. Format: The first word is the file name of the executable, the second part is an optional parameter that specifies which registry keys to use for this instance.

23.2 TC/PreProcess

TC/PreProcess uses the following registry values within the sub key **TCPrep**. The key and the values will be created by TC/PreProcess if they do not exist. The registry values for MetaMail configuration are created empty by TC/PreProcess and must be set in the KCS setup or manually. Values to disable features like DoMetaMail are not created at all and must be created and set manually.

The registry values are read by TC/PreProcess every Watch Thread poll interval which is 300 seconds. Registry values marked with an asterisk ("*") are also read on each call of the processor (every time when a message is played).

Registry Key	Type	Default	Comment
Tracelevel *	REG_DW	0	Enables trace for: 0x00000001 – Non critical errors 0x00000002 – General information 0x00000004 – Normal function call trace 0x00000008 – Full function call trace (could be huge) 0x00000010 – Data trace 0x000000100 – Replacer trace 0x00000200 – Converter trace 0x00000400 – MetaMail resolver trace 0x00010000 – Watch thread trace Levels can be combined.
DoReplace *	REG_DW	1	0 disabled the replacer (tcprep.mpf). This value is not automatically created.
DoConvert *	REG_DW	1	0 disables the converter (tcprep.rul). This value is not automatically created.
DoMetaMail *	REG_DW	1	0 disables MetaMail features. This value is not automatically created.
StopAfterCharacters	REG_DW	10000	Maximum number of characters to be processed
HttpAddress	REG_SZ	<empty>	Name of the server that appears in MetaMail URLs. I.e. "http://webserver/login.asp. "http://" and

			/login.asp" must be contained in the string. This value is compatible to TC/Links value HttpAddress
MMServerName	REG_SZ	<empty>	Name of the server where the messages referred by MetaMail URLs are searched.
MMServerPath	REG_SZ	<empty>	Path of the server where the messages referred by MetaMail URLs are searched.
MMServerUserId	REG_SZ	<empty>	User ID for accessing the MMServer
MMServerUserPwd	REG_SZ	<empty>	Password for accessing the MMServer
MMArchServerName	REG_SZ	<empty>	Same as MMServerName but for archive server (optional)
MMArchServerPath	REG_SZ	<empty>	Same as MMServerPath but for archive server (optional)
MMArchServerUserId	REG_SZ	<empty>	Same as MMUserId but for archive server (optional)
MMArchServerUserPwd	REG_SZ	<empty>	Same as MMUserPwd but for archive server (optional)
PreURLText	REG_SZ	<empty>	HelpText which TC/At2URL can insert before MetaMail URLs. This text is removed by TC/PreProcess.
PreURLTextForTermination	REG_SZ	<empty>	HelpText which TC/At2URL can insert before MetaMail URLs. This text is removed by TC/PreProcess.

Table 27 – TC/PreProcess Configuration Parameters

23.3 TCE_RES

Registry Key	Type	Default	Comment
TCRT\ResFlushInterval	DWORD	5	The interval (in seconds) where the resource engine will flush the record buffer to TCOSS. This only applies if the recording is to be saved on TCOSS.
TCRT\TTSConvertSpeed	DWORD	0xFF00	Most TTS Engines allow to specify how fast to convert text to speech. The default value indicates: "As fast as possible"
TCRT\TTSSBufferSize	DWORD	5120	[kByte] The maximum size of an internal buffer the tce_res reserves when converting text to speech. This would be ~5.3 minutes speech.
TCRT\TTSSentenceLength	DWORD	128	Max number of characters in a sentence. The tce_res will split a sentence with more than this number of chars. This setting allows faster conversion/feedback because of lesser text to convert at once
TCRT\MaxTextLength	DWORD	5000	Every text that is longer than 5000 chars will be trimmed. This means that the tce_res will only speak up to the 5000 first chars of a text
TCRT\TTSKeepTempFiles	DWORD	0	If set to "1", temporary files created for TTS conversion will not be deleted.
TCRT\SendDelay	DWORD	0	Delays the sending of audio data for X msec.
TCRT\ResCacheInitSize	DWORD	2551	The initial size of the cache hash table. For performance reasons, this should be a prime number.

Table 28 – TCE_Res Configuration Parameters

23.4 TCE_Rem

Note that the following registry keys are located under “**HKLM\Software\TOPCALL**” and **not** under “**HKLM\Software\TOPCALL\TCECP**”

Registry Key	Type	Default	Comment
RemoteEngine\UseRPCLocator	DWORD	0	The RPC locator is a specific Windows NT service (RPCLocator) using the standard network browser to find all machines connected to the network. The browser is an integral part of the Windows network services and requires them to be installed and activated. For networks where Windows networking must not be used, the use of the locator can be disabled.
RemoteEngine\UseBroadcastLocator	DWORD	1	An alternative to finding nodes using the RPC locator is to broadcast queries into the network. These queries are proprietary for the remote engine. To prevent them from causing an excessive amount of undirected network packets, a node sends a maximum of two queries at start-up and only one during shutdown. A broadcast is a very quick way to find servers within the network. The use of broadcast can be disabled
RemoteEngine\RpcTimeout	DWORD	50000	Due to a long waiting time that can occur when a connection is established, a timeout can be set. In some cases, there could be a timeout (from the TC/Player,...) during a connection. By reducing the value of this key, this can be solved.
RemoteEngine\DisableBrowser	DWORD	0	The browsing mechanism can be totally disabled. If disabled, the internal database will not be updated with information about public nodes using mechanisms like RPC locator, broadcasts, cache, WellKnownServer, local endpoint map, ... The exact behavior of this switch is not documented may change in future versions.
RemoteEngine\Protocols	MULTI_SZ	"ncalrpc" "ncacn_ip_tcp" "ncacn_spx"	The value is a number of strings representing valid RPC protocol sequences (case sensitive). Only protocols contained within this set will be used even if the local network installation provided more. Protocols specified in this registry value but not provided by the system are silently ignored and do not cause any error as long as at least one is valid. Protocols are used in the order specified. Protocols with higher precedence are specified first.
RemoteEngine\BroadcastProtocols	MULTI_SZ	"ncadg_ip_udp" "ncadg_ipx"	For broadcasting a different list of protocols is used. Protocols specified for broadcasting may overlap with protocols used for standard transmission. Any specified but non-working protocols are silently ignored. The default value is appropriate for TCP/IP and IPX/SPX

			systems.
RemoteEngine\ComTimeout	DWORD	1	The level of tolerance against slow peers i.e. the timeout aggressiveness, can be specified. Valid values are from 0 to 10. A Value of 0 means low tolerance - highly aggressive behavior, and a value of 10 means highest tolerance - no timeout at all. This value is an essential factor for browsing speed. It should not be changed.
RemoteEngine\ChannelTimeout	DWORD	0 [seconds]	A channel may be inactive only for a certain amount of time or it will be closed. This should prevent from message flows being stuck and the need to forcefully terminate applications. 0 deactivates the channel monitoring
RemoteEngine\MaxChannels	DWORD	200	The maximum number of channels being opened at the same time is restricted. If this value is reached the remote engine denies any attempt to open a new connection
RemoteEngine\CacheLifeTime	DWORD	2880 [minutes]	Specifies the time in minutes an entry can be invalid until it is removed.
RemoteEngine\PrivateNodeLifeTime	DWORD	15 [minutes]	The remote engine may keep private nodes temporarily in the internal nodes database. Unlike public nodes, private nodes automatically expire and disappear after a certain period of time.
...\\TCECP\\TCRT\\REMTYPE	DWORD	1	The type of application (remote engine node). Possible Values are 0 (Private) and 1 (Public). Server applications must have this value set to 1. A value of 1 causes publication of the node and its resources.

Table 29 – TCE_REM Configuration Parameters

24. Appendix A – Index

- Architecture
 - Fault tolerance, 26
 - Greeting groups and scheduler, 116
 - Layers, 12, 94, 95
 - Load balancing, 27, 28
 - Overview, 12
 - Service provider interface, 30
 - TC/VoiceAccess, 31
 - TC/VoiceLink2, 37
 - TC/VoiceXML, 34
 - Voice Server, 12, 13
 - Audio caching, 94
 - Audio formats, 178
- Call Flow
 - Get/Set a greeting, 118, 119
 - Preprocessor, 33
 - TC/VoiceAccess, 32
 - Voice Server, 13, 14
- Call Transfer
 - Blind, 18
 - Bridged, 17
 - Loop detection, 18
 - tromboning, 17
- Configuration
 - General, 77, 86
 - Plug-Ins, 78
 - Resource engine, 202
 - Service provider interface, 77
 - TC/PreProcess, 128, 201
 - Converter, 134
 - MetaMail resolver, 130
 - Replacer, 132
 - TC/VoiceAccess, 79
 - Account locking, 125
 - Alternative operator, 105
 - Alternative telephone number and representative, 122
 - Attendant, 101
 - Call sender, 106
 - Customized greetings and system prompts, 102
 - Customized subject, 125
 - Dial by name and distribution lists, 110
 - Different languages for different access numbers, 104
 - E-Mail reader support, 107
 - Fax with voice, 108
 - Greeting scheduler and greeting group, 115
 - Greeting wizard, 123
 - Integration with Kofax Communication Server Archive, 127
 - Lock filter, 124
 - Mailbox sort order, 113
 - Password expiry, 126
 - TCSI codepage conversion, 123
- User depending languages, 103
- TC/VoiceLink2**
 - Customizing Audio Extensions, 163
 - Enhanced message termination, 152
 - Number Locking, 161
 - Poll cycle, 149
 - Remote voice server**, 146
 - TCSI codepage conversion, 148
 - Text to speech, 154
 - Text2Wave Pre-Conversion, 162
- TC/VoiceXML
 - built-in grammars, 136
 - Defaults, 137
- TC/VRedirect
 - Service of a voice call, 164
- TCECP, 86
 - Delayed active Plug-in start, 85
 - Maximum memory usage, 85
 - Performance counters, 86
 - Timeouts, 86
 - TRS, 85
- TCRT, 93
 - Dynamic reconfiguration, 88
 - Graceful shutdown, 87
 - Hints, 189
 - KCS Text to speech configuration tool, 90, 96, 99
 - Microsoft Text-to-Speech API, 89
- Plug-Ins
 - Active and passive, 15
 - Concept, 14
 - Interface, 15, 78
- Requirements
 - External call transfer, 18
 - Hardware and Software, 44
 - ISDN, 44
 - LAN bandwidth, 47
 - Service provider interface, 46
 - TC/PreProcess, 45
 - TC/VoiceAccess, 44
 - TC/VoiceLink2, 45
 - TCRT, 94
 - Voice Channels, 45
 - Voice over IP, 44
- Restrictions, 184
- Scenarios, 165, 166, 168
- Scripting
 - Transparent routing script, 16
- Service provider interface, 30
- Speech Syntheses Markup Language, 96
- TC/Mon32, 84
- TC/UserTool, 119
- TC/VoiceAccess, 10

- Auto login, 81
- Message pre-processing and MetaMail support, 32
- TC services, 79
- TC/VoiceLink2, 10, 34, 168
- TC/VoiceXML, 10, 33
- TCOSS
 - TC/PreProcess, 82
 - TC/VoiceAccess, 80, 82
 - TC/VoiceLink2, 83
- Text to speech, 180
 - Profiles, 29
 - Upgrade/Downgrade, 193
- Tracing
 - TC/PreProcess, 129
 - TC/VoiceAccess, 126
 - TC/VoiceLink2, 155
 - TC/VoiceXML, 136
 - TC/VRedirect, 164
 - TCDownloadFile, 94
 - TCECP, 88
 - TCRT, 93
- Troubleshooting, 187
- UC0 Channel, 79
- Voice Server
 - Group, 23, 100
 - Installation, 52, 57
 - Prerequisites
 - Text to speech, 56
 - Pre-Requisites
 - Service provider interface, 52
 - Restrictions, 52
 - Service provider interface, 66, 69, 174
 - TC/PreProcess, MetMail, 62
 - TC/VoiceAccess, 60
 - Telephone system, 70, 74, 76
 - VoiceLib, 70
 - Languages, 91
 - Licensing, 83
 - Performance, 86
 - Remote connection, 23
- VoiceXSP, 43, 166

25. Appendix B – Tables and Figures

25.1 Tables

<i>Table 1 – Answering Machine Detection Test Results</i>	21
<i>Table 2 – Answering Machine Detection Configuration Parameters</i>	22
<i>Table 3 – Answering Machine Detection Frequencies and Durations.....</i>	23
<i>Table 4 – VL2 Message Types.....</i>	40
<i>Table 5 - Number of Voice Channels</i>	45
<i>Table 6 – TCOSS Services Used by VA.....</i>	80
<i>Table 7 – TC Languages</i>	91
<i>Table 8 – TCRT Tracing</i>	93
<i>Table 9 – Valid Transport Protocols for Prompt Playback</i>	94
<i>Table 10 – Customized Greetings and System Prompts.....</i>	103
<i>Table 11 – Greeting Flags</i>	116
<i>Table 12 – Scheduler Flags.....</i>	117
<i>Table 13 – TC/PreProcessor Wildcards</i>	134
<i>Table 14 – VL2 Error Codes.....</i>	151
<i>Table 15 – TCRT Error Codes.....</i>	153
<i>Table 16 – TCSI Break Codes.....</i>	154
<i>Table 17 – VL Action Abbreviations</i>	154
<i>Table 18 – VL2 Trace Levels</i>	155
<i>Table 19 – TC/VRedirect Trace Levels</i>	164
<i>Table 20 – Supported Audio Codec's.....</i>	178
<i>Table 21 – Supported TTS Engines.....</i>	180
<i>Table 22 – Performance Data.....</i>	183
<i>Table 23 – Documentation Changes 7.80.00ed5 – 7.80.04.....</i>	195
<i>Table 24 – Third Party Licenses</i>	196
<i>Table 25 – Abbreviations</i>	198
<i>Table 26 – Glossary</i>	200
<i>Table 27 – TC/PreProcess Configuration Parameters</i>	202
<i>Table 28 – TCE_Res Configuration Parameters</i>	202
<i>Table 29 – TCE_Rem Configuration Parameters</i>	205

25.2 Figures

<i>Figure 1 – The Three Layer Architecture</i>	12
<i>Figure 2 – The Voice Architecture in Details</i>	13
<i>Figure 3 – A Simple Call Flow</i>	14
<i>Figure 4 – Internal Call Transfer</i>	17
<i>Figure 5 – External Call Transfer Prior to the Transfer</i>	18
<i>Figure 6 – External Call Transfer After the Transfer</i>	18
<i>Figure 7 – Call Transfer Loop Detection.....</i>	19
<i>Figure 8 – Multiple Voice Server Groups.....</i>	24
<i>Figure 9 – System in Full Fault Tolerant State. Both Voice Servers Are Running</i>	26
<i>Figure 10 – Primary Voice Server Became Unavailable (i.e. Power Off)</i>	27
<i>Figure 11 – Load Balancing Based on Resources</i>	28
<i>Figure 12 – Fault Tolerance Based on Resources</i>	28
<i>Figure 13 – Inbound call distribution</i>	28
<i>Figure 14 – TC/MC Speech Profiles</i>	30
<i>Figure 15 – SPI Architecture</i>	30

Figure 16 – VA General Architecture	31
Figure 17 – VA – Typical Call Flow	32
Figure 18 – Pre-Processor Workflow	33
Figure 19 – VoiceXML Architecture	34
Figure 20 – VL2 Architecture	35
Figure 21 – VL2 Architecture and Modules	36
Figure 22 – VL2 on a VS with Telephone System Integration	37
Figure 23 – VL2 Connecting to a Secondary VS	37
Figure 24 – VL2 Polling Concept	38
Figure 25 – VoiceXSP Concept	43
Figure 26 – First Setup Screen	58
Figure 27 – Sample VoiceXSP Instance Selection	59
Figure 28 – TC/VoiceXML – TC/VoiceAccess Warning	59
Figure 29 – VoiceAccess Basic Parameters 1	60
Figure 30 – VoiceAccess Basic Parameters 2	61
Figure 31 – PreProcessor Parameters 1	62
Figure 32 – PreProcessor Parameters 2	63
Figure 33 – VoiceAccess Extended Parameters	64
Figure 34 – VoiceAccess Trace Options	65
Figure 35 – Exchange SPI 1	66
Figure 36 – Exchange SPI Information	68
Figure 37 – Exchange SPI Message Classes	68
Figure 38 – Exchange SPI Custom Message Classes	68
Figure 39 – Lotus Notes SPI Parameters	69
Figure 41 – Install DrWatson for TC/VoiceAccess	69
Figure 42 – Successful Installation of DrWatson for TC/VoiceAccess	70
Figure 43 – Install the VoiceLib	70
Figure 44 – Choose the Telephone System for TC/VoiceAccess	70
Figure 45 – vmaildep.bat	71
Figure 46 – VL2 Group Selection	71
Figure 47 – VL2 Instance Selection	72
Figure 48 – VL2 Upgrade Warning	72
Figure 49 – VL2 Instances	73
Figure 50 – VL2 Licensing	73
Figure 51 – Install DrWatson for VL2	73
Figure 52 – Successful DrWatson installation for the VL2	74
Figure 53 – Vlinkdep.bat	74
Figure 54 – Choose telephone system for TC/VoiceLink2	74
Figure 55 – VL2 configure TTS dialog	74
Figure 56 – TC/VoiceXML Licensing	75
Figure 57 – Install DrWatson for TC/VoiceXML	75
Figure 58 – DrWatson Installed Successfully for TC/VoiceXML	76
Figure 59 – Choose Telephone System for TC/VoiceXML	76
Figure 60 – User VL Rights	83
Figure 61 – TC/Mon32 Output for VA	84
Figure 62 – TC/Mon32 Output for VL2	84
Figure 63 – TC/Mon32 Graceful Shutdown	87
Figure 64 – TC/Mon32 Dynamic Reconfiguration	88
Figure 65 – TTSCConf Startup Screen	90
Figure 66 – TTSCConf language	92
Figure 67 – Audio Cache Layers	95
Figure 68 – SSML Prefix with TTSCConf	97
Figure 69 – Force XML for SSML	98
Figure 70 – SAPI 4 GUID	99
Figure 71 – SAPI 5 GUID	100
Figure 72 – Choose a Language in TCFW	104

<i>Figure 73 – TCSI – Tree for Greeting Groups and Scheduler.....</i>	116
<i>Figure 74 – VA Flowchart to Get a Greeting</i>	118
<i>Figure 75 – VA Flowchart to Set a Greeting</i>	119
<i>Figure 76 – TCfW Number Locking.....</i>	162
<i>Figure 77 – A Typical Scenario</i>	166
<i>Figure 78 – Voice xSP Scenario Setting</i>	167
<i>Figure 79 – VoiceXML Scenario Setting.....</i>	168
<i>Figure 80 – Phone Emulator Dialog.....</i>	174
<i>Figure 81 – Performance Scenario.....</i>	182