Telephony Capture Service

User Manual

Version



**Document Change History**

Table of Contents

1 Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Document Location 4

1.4 Document Status 4

1.5 Acronyms Definitions 4

1.6 References and Related Documents 4

1.7 Open Issues 4

2 Preliminaries 5

2.1 Assumed User Characteristics 5

2.2 Maintaining Environment Variables 5

3 Use Cases 6

3.1 Launching the TCS 6

3.2 TCS Health Monitoring 7

3.3 Rolling Back to a Previous TCS Version 7

3.4 TCS Software Upgrades 7

3.5 Restoring a Database Backup 8

3.6 Management of Database Backups and xLogs 8

# Introduction

## Purpose

The purpose of this document is to provide specific instructions for how to carry out the various TCS use cases.

## Scope

This manual details how to start, stop, upgrade, and monitor the performance of the TCS. And because the TCS has the further responsibility to install and otherwise exploit a new database, it also includes instructions for how to restore the database from a backup set.

This document does not, however, include instructions for how to carry out any of the following:

* The various admin management functions associated with running a Linux installation.
* Instructions for installing docker, docker-compose, nor Kitematic.
* Instructions for how to modify the TCS database, the queuing service, nor the custom TCS software itself. There is a separate manual for such things, namely the TCS Development Manual.

## Document Location

This document is found in the ‘docs’ folder of the TCS GitHub repository (refer to the TSC Software Requirements Document for the url of the GitHub repository).

## Document Status

This document must be kept current and released concurrently with each software release.

## Acronyms Definitions

The reader is referred to the TCS SRD.

## References and Related Documents

The reader is referred to the TCS SRD.

## Open Issues

None

# Preliminaries

## Assumed User Characteristics

## Maintaining Environment Variables

A number of TCS-specific environment variables must be set in the following file:

~/.tsc.bash

As this file contains confidential information, it is not maintained in GitHub.

The SRD contains detailed descriptions of the various variables. Failure to set all the environment variables to appropriate values will certainly result in the loss of one or more TCS features.

## tcsproj

# Use Cases

## Launching the TCS

Proceed as follows:

1. cd ~
2. rm -rf tcs/\*
3. git clone --branch <tcs-version><https://github.com/roderickmonk/telephony-capture-service.git> ~/tcs  
   (This actually gets everything: code, docs, Docker config files, etc. I will codify this beast into a script for you, so that you don't have to worry about its messiness).
4. cd ~/tcs
5. tcsproj  
   This script sets up a number of aliases, exports, and creates some bash functions.  In particular, it will activate those environment variables that discussed above.  Also, if the user opens any more terminals, you can again call tcsproj in each one to make other terminals ready for TCS related activity.

**Scripts for Launching TCS Containers**

* tcp-up  
  This is the big one: it launches all the containers that contain mainstream TCS processing.  These are:
  + pbx-interface
  + tms-interface
  + database-interface
  + tcs-postgres (a tcs specialization of the postgres database).
  + rabbitmq
  + backup-scheduler

Key point: TCS\_VERSION must be set to the required value.

Note that NO building of software is required.  This is because tcp-up will reach out to Docker Hub and, if not found locally, will download a Docker image ccbcadmin/tcs-image:v1.3.  A developer would have already put it there ready to be downloaded and put to use.  This image contains installs of postgres, Node, and the TCS software.

* Tms-simulator-up  
  During system / acceptance testing someone may want have a sink for data destined for the TMS (otherwise the queues will grow without limit, if there is an input PBX source.  Note that if you have a need for tms-simulator-up, then you are probably also running pbx-simulator as well - see below).

Thus far all of the containers started by tcs-up and tms-simulator-up run in the background as daemons.  This is not the case for the next two.  These run in the forward, that is, they do not release the session until they complete.

* pbx-simulator source-smdr-directory  
  (e.g. pbx-simulator /smdr-data/smdr-data-002)
* This simulator opens a circuit to the container pbx-interface and sends SMDR messages in chronological order drawn from the data in the specified directory).
* mangle source-smdr-directory target-smdr-directory  
  (e.g. mangle /smdr-data/smdr-data-002 /smdr-data/smdr-data-003)

## TCS Health Monitoring

Clearly, you are going to want to know how the TCS is doing.  This is my current thinking on the subject:

* Kitematic: I think you already have an idea about Kitematic can provide in the way of monitoring.  You will be able to ensure (or not) that all the containers are working, etc.
* RabbitMQ management console: there is a wealth of features provided by this web portal, but the important one in this context is the ability to view how many messages are in the queues (nominally the queue size should be more or less 0 if everything is working properly).
* Logs: I have some work to do on this subject, but my intention is to gather up all the process logs from the various containers and funnel them all to a single channel from which you can study them after the fact ("e.g. something strange happened last Thursday - maybe the logs recorded a problem").

## Rolling Back to a Previous TCS Version

1. cd ~/tcs; tcs-down
2. This will shut down all things TCS.
3. Change the environment variable TCS\_VERSION in the file of environment variables.
4. tcs-up
5. Note: the image for the rollback version MAY still be available locally, in which case no pull from the Docker Hub will be required and hence the TCS will be again be active that much faster.  Either way, the TCS will again become active, but this time running the software as it existed in the rollback version.

## TCS Software Upgrades

This use case is completely covered by the section **Launching the TCS Containers** (i.e. starting afresh is the same as doing an upgrade).

## Restoring a Database Backup

## Management of Database Backups and xLogs