TCSMGR

TCS management software

User and Developer Manual

Project name TCS Replacement Project

Release Draft: Version 0.1
Date: 2019-12-13

Author(s):	Jure Skvarč
Owner:	Jure Skvarč
Client:	ING
Document Number:	ING-TCSR-COM-XX

Document History

Document Location	The document can be found at : http://www.ing.iac.es/bscw/bscw.cgi/XXXXXX

Revision History

Revision date	Version	Summary of Changes	Changes marked

Approvals This document requires the following approvals.

Name	Title	Approval Date	Issue Date	Version

Distribution This document has been distributed to:

Name	Title	Issue Date	Version

Table of Contents

1 Applicable and Referenced Documents....5
1.1 Applicable Documents....5
1.2 Referenced Documents....5
2 Abbreviations and glossary....5
3 User manual....6
3.1 Purpose....6
3.2 Architecture....6
3.2.1 Server....6
3.2.2 Client....6
3.2.3 User roles....7
3.3 Starting the server....7
3.4 Starting the GUI....7
3.5 GUI description....8
3.5.1 Status panel....8
3.5.2 Response panel....8
3.5.3 Command line....9
3.6 User sessions....9
3.7 Process monitoring....9
3.8 Managed items....9
3.9 Commands....10
3.9.1 user <user name>....10
3.9.2 exit....10
3.9.3 start <application name>....10
3.9.4 restart <application name>....11
3.9.5 stop <application name>....11
3.9.6 show <resource> <application name>....11
3.9.7 edit <resource> <application name>....11
3.9.8 log....12
3.9.9 versions <application name> | all....12
3.9.10 set....12
3.9.11 deploy <application name> <version>....12
3.9.12 help [<command>]....12
3.9.13 upload <application name> <versions>....13
3.9.14 remove <application name> <version>....13
3.9.15 reboot....13
4 Use cases....13
4.1.1 Get an account....13
4.1.2 Monitoring current status....13
4.1.3 Logging in an out....13
4.1.4 Starting and stopping of applications....14
4.1.5 Uploading and deploying new software versions....14
5 Developer's and maintainer's manual....15
5.1 Hardware....15
5.2 Software dependencies....15
5.2.1 Languages....15
5.2.2 External libraries....15
5.2.3 Services....15
5.2.4 Operating system....15
5.3 Source file structure....15

5.4 Adding new users....16
5.5 Source code documentation....16
5.5.1 Coding and documenting guidelines....16
5.5.2 Tools....16
5.5.3 Configuration....17
5.5.4 Creation....17
5.6 Version control....17
5.7 Deployment....17
5.7.1 tcsmgrd....17
5.7.2 Computers....18
5.8 Application details....19
5.8.1 Tcsmgrd....19
5.8.2 Client....19
5.8.3 Communication protocol....19
5.8.4 Message types and formats....19
6 Configuration files and parameters....21
6.1 Tcsmgrd....21
6.2 Tcsmgr....21
7 Managed files....21
7.1 Applications....21
7.2 Configuration files....22
8 Log files....23
9 Troubleshooting....23
9.1 Tcsmgrd....23
9.2 Tcsmgr....24

1 Applicable and Referenced Documents

1.1 Applicable Documents

Applicable documents are those documents whose content is considered to form part of this document. The specified parts of the applicable documents carry the same weight as if they were stated within the body of this document. The applicable documents are:

Document Identifier	Document Title

1.2 Referenced Documents

Document Identifier	Document Title
ING-TCS-SOFT-01	TCS Bridge Developer and User Manual (https://bscw.ing.iac.es/bscw/bscw.cgi/1057615)
ING-TCS-UTIL-01	TCSPLOT - Telemetry extraction and Plotting Tool for WHT TCS

2 Abbreviations and glossary

MQTT	A simple messaging protocol (http://mqtt.org)
TCS	Telescope control system
TCS machine	A general term for a machine on which the TCS software is running. This does NOT refer to the VMS machine.
tcsbridge	The operational machine on which the managed TCS processes are running.
tcsdev	Development machine for tcs_bridge application
tcsdevspare	Spare machine for tcsdev
WHT	William Herschel Telescope

3 User manual

3.1 Purpose

The new TCS for the WHT has a number applications running on a Linux PC which need to be managed in an efficient an centralised manner. Tcsmgr is a software system used to monitor, install, deploy, configure and generally manage all TCS-related software from one place. Programs which fall into this category are tcs_bridge, tcs_simulator, tcsplotserver, tailserver, weaveFts, weaveAdc and tcsmgrd. This list will change as the TCS evolves. The programs are running on tcsbridge computer. It is expected that with the final development of the TCS on Linux the new TCS will run on the same machine.

The advantages of using a special interface program are:

- We avoid ssh sessions on tcsbridge which are potentially a security risk and can be time-consuming to execute.
- We limit the number of available commands which are normally available to the root user, reducing the likelihood of confusion.
- Help on TCS management commands is available within the application.
- All actions are logged.
- Some actions would be awkward to perform manually and would likely be applied inconsistently when executed. Tcsmgr simplifies this.

The existence of TCS manager does not eliminate completely the need to still perform many tasks in interactive terminal session and instead focuses on the tasks which are performed frequently and routinely.

3.2 Architecture

Tcsmgr is implemented in a client-server architecture, allowing several simultaneous client sessions. While it is clearly not desired that multiple users make changes to the system at the same time, it is on the other hand convenient to allow system monitoring task to run possibly on several computers. The probability of problems caused by several people using possibly conflicting commands is further reduced by having different roles for users and by the fact that in practice there will not be many users who will want make major changes in the system.

3.2.1 Server

The server runs on the TCS machine. It is a Python program called tcsmgrd which is started automatically at the boot time. It assumes the a MQTT server is also running on the tcs machine. Tcsmgrd subscribes to several MQTT topics over which it receives commands from the clients. It also periodically publishes the status of the tcs machine and the managed programs. All received commands and executed actions are logged in a log file /var/log/tcs/tcsmgrd.log.

3.2.2 Client

The client is a Python application which uses a GUI to show system status, accept user commands and show responses to the commands. Several client programs can be running at once, even on the same machine and by the same user.

As in the case of the server, the commands and other important process parameters are logged. Due to the possible multiplicity of the client processes on one machine, the log files have different names and are stored /tmp/tcsmgr directory

3.2.3 User roles

To use the tcsmgr GUI for more than monitoring the status, the user must first log in. It is not desirable to allow all users access to all commands, so each user is assigned a role which determines which commands can be executed. Three user roles are:

Admin: Can execute all commands

User: Can restart applications and reboot the computer

Voyeur: Can only execute read-only commands

3.3 Starting the server

The server executable tcsmgrd must be started using service management framework via the systemctl command. Normally this will happen when the tcs machine boots. If for some reason this didn't happen, it is possible to try to start manually as root on the TCS machine using command `systemctl start tcsmgrd`.

3.4 Starting the GUI

The GUI is installed as a part of the observing system on ICS machines and, where needed, on machines belonging to individuals. Any machine on which the GUI is running must be on the list of machines which are allowed to access the TCS machine. The program is invoked as:

```
tcsmgr
```

This will start the GUI which will try to connect to the MQTT server on machine *tcsbridge*. If we want to connect to the server running on another computer, the program must be started as

```
tcsmgr --host <machine>
```

Possible machine names are also *tcsdev* nad *tcsdevspare*.

3.5 GUI description

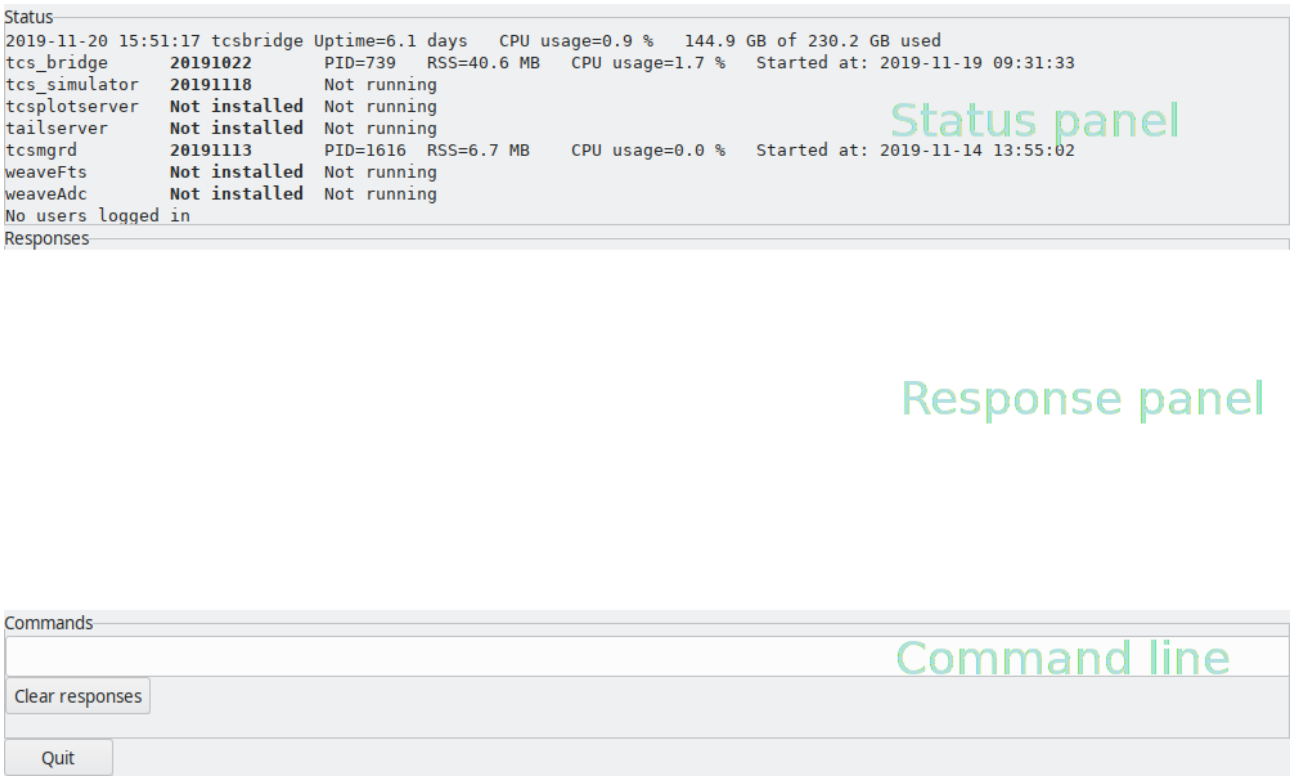

Figure 1: Tcsmgr GUI

The GUI is divided into three main parts: the status panel, the response panel and the command line.

3.5.1 Status panel

The status panel shows dynamically basic information about the tcs machine and the processes relevant for the TCS.

The first line shows the current system time, name of the TCS machine, the uptime, current total CPU usage, and the disk usage.

Next lines are showing the status of managed applications. More details about this are in section 3.7.

Finally, the last line contains the list of active users. The format is

```
username (machine name – PID)
```

The user is identified by the name, by machine name and by the PID of the client process. This allows to unambigiously identify multiple clients on the same computer, even if the same user is logged in in each client.

3.5.2 Response panel

The response panel shows commands that have been issued by the user in blue colour, the responses from the server in black and the errors in red. It is possible to scroll back using the scrollbar slider to see the responses which have scrolled out of the view. The responses can be cleared by pressing the *Clear responses* button, which is located under the command line.

3.5.3 Command line

The commands are entered in the command line. They are sent to the server when the user presses the *Return* key. Usual navigation with the cursor keys is possible. In addition, it is possible to recall up to 1000 previous commands by pressing up and down cursor keys. The history is preserved between logins. For example if the session is the terminated by command *exit* or due to inactivity, the previous commands remain in history. The history is NOT preserved between invocations of the GUI.

3.6 User sessions

To use *tcsmgr* for more than just monitoring of processes, the user must log in. This is done by command

```
user <username>
```

The program will ask for the password. If the entered password is correct, the user is authenticated and can use commands allowed for the role. The session ends when the user executes command *exit* or when the session expires after an hour of inactivity.

3.7 Process monitoring

The first line in the status panel is followed by the status of all managed applications. The application name is followed by the version of the application. If the application is not installed in /usr/local/bin, it will be indicated so (**Not installed)**. There is also a possibility that the application has been installed manually and that the file is an actual executable instead of being a link to the currently deployed version. In this case "**No version"** will be shown instead of the version. Following is the dynamic status information for individual applications: PID (the process number), RSS (the used RAM), CPU usage (in percent of one core capacity) and the application start time.

A significant increase in memory or CPU usage might mean an application error which needs to be investigated, so it is useful to check these values if the TCS starts to misbehave.

3.8 Managed items

Tcsmgr manages a limited set of applications and their configuration files which are related to the TCS. Current list (as of December 2019) is shown in Table 1.

Table 1: Applications managed by tcsmgr.

Application	Status	Description
tcs_bridge	Active	A program passing data between the TCS and the mechanism controllers
tcs_simulator	Active for testing purposes	A limited simulator of the TCS
tcsplotserver	Planned	Server providing data for telemetry plotting
tailserver	Planned	Program for monitoring log files
weaveFts	Active when WEAVE rotator is available	Interface between tcs_bridge and WEAVE FTS PLC controller

Application	Status	Description
weaveAdc	Planned	Interface between tcs_bridge and WEAVE ADC PLC controller
tcsmgrd	Active	Server for the TCS manager

3.9 Commands

The user communicates with the server by entering commands on the command line and pressing return. Commands are checked for correctnes at the server side and the server response is shown in the response panel. Some commands are only allowed when the user has a particular role. The commands are case-sensitive.

3.9.1 user <user name>

Allows user identification. The user must provide a user name from the user database to be able to log in. After this command is entered the user is asked to authenticate by entering a password. On success the user will be able to execute commands allowed by the user role - one of "admin", "user" and "voyeur".
If there is a need to log in as another user from the same GUI session, the current user must first exit and then use this command.
The session is active until the user types *exit* or until 1 hour of inactivity, meaning no commands being sent to the server.
The user must be entered to the password file manually by the administrator of the system.

Roles: None (must *not* be logged in when using this command)

3.9.2 exit

Exit user session. After exiting only status display and commands "user" and "help" are available. The command line history and the response panel are not affected by this command.

Roles: All

3.9.3 start <application name>

Starts given application. After the successful start the application the status panel should show the PID and other process information. Only applications listed in the status panel can be started. This command obviously does not work for tcsmgrd because it needs to run in the first place to have the manager working.
This command requires that a configuration file *<application name>.service* is installed in /etc/systemd/system on the TCS machine.

Roles: admin, user

3.9.4 restart <application name>

Stops and starts again given application. If the application is not running before the command is given, it will be started anyway. In at most few seconds after the successful start the application status panel should show the process PID and other information. Only applications listed in the status panel can be restarted. This command does not work for tcsmgrd.
This command requires that a configuration file *<application name>.service* is installed in /etc/systemd/system on the TCS machine.

Roles: admin, user

3.9.5 stop <application name>

Stops given application. After stopping the application successfuly the status panel should show the status as "Not running". Only applications listed in the status panel can be stopped. Tcsmgrd can be stopped using the GUI but not started or restarted. After the tcsmgrd is stopped no further communication can take place, so it needs to be restarted manually by logging into the TCS machine.
This command requires that a configuration file *<application name>.service* is installed in /etc/systemd/system on the TCS machine.

Roles: admin, user

3.9.6 show <resource> <application name>

Show the given resource.

Possible resources are shown in the table:

Resource	Description
config	Configuration file for the application. The file can be viewed, but not edited
version	Show the currently installed version of the application

Roles: admin, user

3.9.7 edit <resource> <application name>

Show and allow to edit the given resource.

Possible resources are shown in the table:

Resource	Description
config	Configuration file for the application. The file can be edited in the pop-up editor after the "Edit" button is pressed. To save the changes, press the "Save" button. The tcsmgrd on the TCS machine will make a backup copy

Resource	Description
	of the existing configuration file before saving the changes. The old versions of the configuration file are saved with the names <application name>.conf.<n>, where <n> is the number of the backup copy. There can be up to 100 backup copies for each configuration file, with <n> between 1 and 100. The process of restoring the file is manual and must be executed on the TCS machine. Up to 100 backup copies can exist. The backup file with the highest number is not necessarily the most recent file, so attention must be paid to the actual creation time. Old and unneeded backup files must be cleaned up manually on the TCS machine.

Roles: admin

3.9.8 log

Not implemented yet

3.9.9 versions <application name> | all

Shows available versions for the given application or all of them. Different application versions are distinguished by an extension following the application name, for example tcs_bridge.20191112. Typically, dates are used to name different versions, but the version names can be arbitrary strings.

Roles: all

3.9.10 set

Not implemented yet.

3.9.11 deploy <application name> <version>

Make the given version of the application active. This is done on the TCS machine by creating a soft link between the raw application name and the specific version. If the application is running at the time of deployment, it will be first stopped, but it will *not* be automatically restarted. This must be done explicitly by command *start <appname>*.

Roles: admin

3.9.12 help [<command>]

When invoked without arguments, it displays all available commands with short descriptions. When a command name is given, a more detailed command description is given.

Roles: All, including when not logged in.

3.9.13 upload <application name> <versions>

Upload the given application to the TCS machine. The applications are stored in /usr/local/bin/versions/<application name> directory. When invoking this command, the GUI will launch a file selector to select the executable file on the local machine. This will consequently be uploaded to the target machine and renamed as <application name>.<version>, for example tcs_bridge.2019112. The freshly uploaded version of the application does *not* become the released version automatically. For this one must use command *deploy* (see section 3.9.11).
It is not possible to overwrite an existing application version. If this is needed, you must first use the remove command to remove the existing version and then upload the new instance of this version.

Roles: admin

3.9.14 remove <application name> <version>

Remove specific version of the application. Removal is permanent and, once removed, the executable cannot be recovered any more.

3.9.15 reboot

Reboots the TCS machine. Before rebooting the user must re-enter the password.

Roles: admin, user

4 Use cases

4.1.1 Get an account

To use TCS manager for more than viewing the status you need an account. You can ask the software group manager or the TCS manager to create an account for you.

4.1.2 Monitoring current status

If you just want to peek into the current TCS status on the TCS (linux) machine, just start tcsmgr on taurus. It is not necessary to log in or even to have an account.

4.1.3 Logging in an out

Before performing any action on the TCS machine you have to be logged in. To log in, use command

user <username>

and enter your password when asked to.

4.1.4 Starting and stopping of applications

To be able to start, stop or restart TCS application, you must first log in as a user who has either *user* or *admin* role. Then use commands *start*, *stop*, or *restart* to start or stop the applications. If this needs to be done during normal operations (not TCS software maintenance) it would normally be followed by a fault report or a message to the TCS manager explaning the reason.

4.1.5 Uploading and deploying new software versions

New software versions can be uploaded and deployed only by the users with the *admin* role.

To upload, use for example

upload tcs_bridge 20191121

A file selector will appear which will let you select the application to be uploaded. It is important not to confuse the executable with another executable, since anything uploaded will be renamed to match the upload command parameters.

5 Developer's and maintainer's manual

5.1 Hardware

Tcsmgr should be able to run on any recent or not so recent computer with x86 architecture. Other processor types have not be tested but should work, too, providing that the software requirements are met.

5.2 Software dependencies

5.2.1 Languages

Tcsmgr needs Python 3.6 or higher Tcsmgrd need Python 3.7 or higher.

5.2.2 External libraries

External libraries are those which are not part of the standard language libraries and have been developed either by external (to the ING) entities or within previous ING projects.

Name	Version	Purpose	Used in
paho	1.3.1	MQTT library	tcsmgrd.py, wxtcsmgr.py
psutil	5.4.2	Python system and process utilites	tcsmgrd.py
wxPython	4.1.0a1	wxPython GUI framework, based on wxWidgets	wxtcsmgr.py

5.2.3 Services

Name	Purpose	Used by
systemd	Service process management, part of the Linux OS	tcsmgrd.py
mosquitto	MQTT broker. Must run on the TCS machine	tcsmgrd.py, wxtcsmgr.py
OSCAR	Observing system configuration manager. Used to deploy the client on observing system machines.	tcsmgr, wxtcsmgr.py

5.2.4 Operating system

Tcsmgr has been developed and is being run on Linux. In principle it could be adapted for other operating systems, as long as the libraries are available.

5.3 Source file structure

File name	Function
createPassword.py	Independent utility used to create password hashes for the password file
helpPages.py	Help text for all commands. Imported by tcsmgrd.py.
tcsmgrd.py	TCS manager daemon, running on the TCS machine
tcsmgr	Bash invocation script for the GUI (wxtcsmgr.py)
wxtcsmgr.py	TCS manager GUI

5.4 Adding new users

New users must be added manually to the password file /usr/local/etc/tcs/tcsmgr.pass. This must be done by the TCS administrator.

The password file structure is:

<user name> <encrypted password> <role>

The items in each line are separated by spaces. To create the encrypted password, run the program create_password.py. It will ask for a password and will output the encrypted version. This needs to be added to the password file after the user name and before the user role.

The "admin" role must be reserved for one or two people only, as it allows deployment and upload of TCS applications.

Only users who may operationally need to restart the TCS applications should have a "user" role. This should be limited to OSAs and software engineers who are not TCS specialists.

The rest of users should have the "voyeur" role.

5.5 Source code documentation

5.5.1 Coding and documenting guidelines

Coding style should follow the usual PEP-8 style guide. The documentation format is that of epydoc.

It is most important to document especially function purposes, all parameters and the return values.

5.5.2 Tools

The source code is using Epydoc tool with its EpyText markup language. Although at the time of writing (December 2019) Epydoc doesn't support Python 3 and seems like an abandoned project, the alternatives tested haven't appeared to be better in an aestethic sense. If in a longer term Epydoc becomes unusable because of obsolescence, alternatives can be considered:

- Pydoctor (https://github.com/twisted/pydoctor). Should be somewhat compatible with Epydoc. In a quick test it failed to generate documentation for some source files, so it probably needs tweaking.
- Sphinx (http://www.sphinx-doc.org/en/master/) uses a different syntax, so the original source files need to be processed to convert format. This can be done using the Pyment tool (https://github.com/dadadel/pyment/blob/master/doc/sphinx/source/pyment.rst). I a test, the conversion from Epydoc to Sphinx didn't require a lot of work, but it wasn't 100 % complete, so some manual tweaking to the resulting source files is needed.

5.5.3 Configuration

The configuration file for Epydoc is `epydoc.conf` and has been adopted from the `whtpy` project.

5.5.4 Creation

To compile source files:

```
epydoc -v *.py
```

```
cd html
./postprocess.py
```

This will create HTML-based documentation with the starting page in html/index.html. The source code documentation is deployed in `/home/docs/public_html/tcs/tcsmgr` directory.

5.6 Version control

Version control is implemented using GIT. Users need to learn general GIT commands, such as cloning, adding authorisation password for the repository and so on.

To get the source file tree, use

```
git clone https://ingbitbucket.ing.iac.es/scm/soft/tcsmgr.git
```

To add a new source file, put it in the source tree and use git add command, then commit. For example, if you want to add a new source file called new.py:

git add new.py

git commit -m "Initial release of new.py"

5.7 Deployment

Tcsmgr consists of two parts:

1. tcsmgrd, the server program running on the TCS machine
2. tcsmgr, the user interface, running on any computer which has allowed access to the TCS (tcsbridge) machine.

5.7.1 tcsmgrd

Tcsmgrd can be installed on the TCS either manually or using tcsmgr itself. The manual installation is needed the first time the program gets installed on the TCS machine.

First we must take care that the program is configured to start up automatically when the machine is started and that it can be started, stopped and enabled using system commands. For this we need tcsmgrd.service file which resides in /etc/systemd/system directory. The file content is shown here:

```
 [Unit]
Description=tcsmgrd service
After=network.target

[Service]
Environment="PYTHONPATH=/usr/local/lib/python3.7/site-packages"
User=root
Group=root
Restart=always
RestartSec=1
Type=simple
WorkingDirectory=/var/log/tcs
ExecStart=/opt/anaconda/bin/python /usr/local/bin/tcsmgrd --
broker=localhost
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
```

The service is enabled by

```
systemctl enable tcs_bridge
```

It is started by

```
systemctl start tcs_bridge
```

and stopped by

```
systemctl stop tcs_bridge
```

5.7.2 Computers

Currently the tcsmgrd server is deployed on the computers summarised in Table 2.

Table 2: List of computers on which tcsmgrd is deployed.

tcsbridge	The operational TCS machine
tcsbridge spare	A spare for the operational TCS machine. It is an exact copy, including the IP address and other settings, therefore it is normally disconnected from the network and not running. Whenever a new tcsmgrd version is deployed on tcsbridge, it must be installed on the spare tcsbridge, too.
tcsdev	The development machine for TCS software. Mostly a copy of tcsbridge. Operationally does not need to run tcsmgrd, but it is used for testing it, amongst other TCS components.

tcsdevspare	A copy of tcsdev. Mostly used for testing. Tcsmgrd must be deployed.

5.8 Application details

As explained in section 3.2, the tcsmgr system consists of two parts: tcsmgrd server, which is running on the TCS machine, and tcsmgr GUI or client, which can run on any machine permitted to communicate with the TCS machine.

5.8.1 Tcsmgrd

Tcsmgrd is constantly running as a service on the TCS machine. It has the following functions:

1. Constantly monitors some of the most important process parameters
2. Communicates status information to client applications
3. Allows client applications to execute commands on the server
4. Manages user sessions by handling user authentication, taking into account user roles and maintaining a list of active users. Long inactive users are automatically disconnected after a time-out period
5. Accepts, verifies and executes user commands and prepares responses and error messages.

Tcsmgrd does not assume much about the client, so different clients are possible, as long as they adhere to the protocol and correctly form the messages. The responses and error messages are using a limited subset of HTML tags, so they are meant to be rendered on the client side using HTML-aware components.

5.8.2 Client

The tcsmgr client is realised in a form of a GUI, implemented using wxPython framework. It subscribes to status messages from the server and to dedicated response topics. The client identifies itself by an ID generated from the machine name and process ID. This allows the server to distinguish possibly several clients

5.8.3 Communication protocol

Tcsmgr is using MQTT to transmit messages between the server and the client. MQTT communications are done through separate communication channels - topics, to which clients can subscribe or can publish to them. All communication is done via broker called mosquitto which, for tcsmgr, must be running on the TCS machine, so that it is local to the server. Unlike in some other protocols, in MQTT there is no distinction is usage between clients and server - both are clients to the broker. The actual role is defined by the application logic.

5.8.4 Message types and formats

When a message arrives to a topic to which an application has subscribed, a callback is called and it is up to application to interpret the message. Different types of messages arrive on different MQTT topics. They are summarised in Table 3.

Table 3: MQTT topics used in tcsmgr software. To get a full topic name, one needs to add string "tcsmgr/" before the topic name, for example "tcsmgr/command". For communications from server to client, client id added at the end, for example "tcsmgr/response/taurus-12345".

Name	Direction	Purpose
command	Client -> server	Send a command from the client to the server
autocommand	Client -> server	Send a response to the server request. Used for commands which are not typed by the user explicitly, such as send password when rebooting and to send configuration file from client back to the server. In retrospective, it could be possible to have only command topic.
response	Server->client	Non-error response to the user command. It is for a specific client, so it needs client id added.
status	Server->client	Send status of the TCS computer and its applications. This is a general broadcast topic, so needs no client id.
error	Server->client	Error message for the client, for example when an non-existent command is received. For a specific client, so it needs client ID. Error messages are shown in red in the GUI.
session	Client -> server	To send commands related to maintaining user session.
request	Server->client	A request from the server to the client to do certain action, for example show a password prompt or upload a file. Needs client id.
file	Client -> server	Send a file from the client to the server

Message formats are summarised in Table 4. Most messages from the client to the server contain the client ID (something like taurus-12345) and the authentication code which was generated on the server and passed to the client, to be used in all messages sent to the server. These two are prepended by the client to any message before sending. Session messages do not contain the authentication code.

Table 4: Message formats

Name	Format
command	<client id> <auth> <command> [arguments...]
autocommand	<client id> <auth> <command> [arguments...]
response	Message (string)
status	A string, consisting of lines separated by "\n" character
error	Error message (string)
session	<client id> <session command> [arguments...]
request	<request command> [arguments...]
file	<client id> <auth> <file content>

6 Configuration files and parameters

6.1 Tcsmgrd

Tcsmgrd does not have a configuration file. It does need a password file for the users. The password file location is `/usr/local/etc/tcs/tcsmgr.pass`.

Since it is permanently running as a service, the command line parameters are only used for debugging purposes. In production the broker is always localhost. The complete list of options:

```
-h, --help          show this help message and exit
--id ID             Client ID - normally just use the default
--broker BROKER     Name of the machine where the broker is running
--logfile LOGFILE   Full path of the log file
--debug             Whether to use debugging messages in the log file
```

6.2 Tcsmgr

Tcsmgr does not have a configuration file. Of the command line options the only which could be used by users is `--broker` (or the alias `--host`). The list of options:

```
-h, --help            show this help message and exit
--id ID               Client ID - normally just use the default
--broker BROKER, --host BROKER
                      Name of the machine where the broker is running
--logdir LOGDIR       Name of the logging directory
--debug               Whether to use debugging messages in the log file
```

7 Managed files

7.1 Applications

All applications reside in /usr/local/bin directory and the directories below. Different versions of applications are in /usr/local/bin/versions directory. There is a dedicated directory for each application, and different version of the actual executables are inside that directory. In /usr/local/bin is a link to the currently deployed executable. An example is given in table 5. Directories are shown on bold, links in italics and files in normal font. In this example, not all applications have actually installed executables.

Table 5: Directory and file structure for application executables.

/usr/local/bin	**versions**	**tcs_bridge**	tcs_bridge.20190920
			tcs_bridge.20191003
			tcs_bridge.20191121
			tcs_bridge.20191122
		tcs_simulator	
		tcsplotserver	
		tailserver	
		tcsmgrd	tcsmgrd.20191113
			tcsmgrd.20191205
		weaveFts	
		weaveAdc	
	tcs_bridge->versions/ tcs_bridge/ tcs_bridge.20191122		
	tcsmgrd->versions/ tcsmgrd/ tcsmgrd.20191205		

7.2 Configuration files

TCS applications may have configuration files, which are stored in subdirectories od /usr/local/etc. The locations relative to /usr/local/etc are shown in Table *6.*

Table 6: Locations of configuration files for TCS applications.

Application	Configuration file location relative to /usr/local/etc
tcs_bridge	tcs_bridge/tcs_bridge.conf
tcs_simulator	tcs/tcs_simulator.conf
tcsplotserver	tcs/tcsplotserver.conf
tailserver	tcs/tailserver.conf
tcsmgrd	tcs/tcsmgrd.conf
weaveFts	tcs_bridge/weave_fts.conf
weaveAdc	tcs_bridge/weave_adc.conf

When a configuraton file is replaced, the old version is renamed by adding a number, for example tcs_bridge.conf.5. Up to 100 backup copies can exist.

8 Log files

Log files are written in a predetermined location, unless another directory is given in the command line. If an error happens during the creation of log files, the output will be redirected to the terminal instead.

Several clients may be running on the same machine simultaneously, so each has its own log file. In theory this could generate a large disk usage on the computer where the client is running. In practice, it is not expected to happen. The /tmp directory should be cleared after every reboot.

Application	Location	Max size (bytes)	Backup count	Comment
tcsmgrd	/var/log/tcs/tcsmgrd.log	10000000	5	Default level is info. Use --debug option for debugging
tcsmgr	/tmp/tcsmgr/tcsmgr-client-<pid>.log	10000000	5	Default level is info. Use --debug option for debugging

9 Troubleshooting

9.1 Tcsmgrd

If tcsmgrd doesn't start on the TCS machine, inspect the system log using `journalctl -x` command. This should show the error messages. For more interactive experience it is recommended to start tcsmgrd from the command line, following PYTHONPATH initialisation and Python interpreter as specified in `/etc/systemd/system/tcsmgrd.service`. Option `--debug` will show more logging messages. You must be root to start tcsmgrd manually.

In some cases it may be useful to run an older version of tcsmgrd, to see if some external circumstance is causing the problem.

9.2 Tcsmgr

If tcsmgr fails to start or has problems during execution, software group should be called for help, as there is nothing that other users can do. Things that can go wrong are listed in Table 7.

Table 7: Failure modes of the tcsmgr program.

Symptom	Possible cause	Solution
Program not starting. Example message: `(tcsmgr: line 12: bin/python3: No such file or directory)`	Missing correct version of Python intepreter.	Check tcsmgr invocation script and and verify that the path to the Python interpreter actually exists.

Symptom	Possible cause	Solution
Program not starting. Example message: `ModuleNotFoundError: No module named 'wx'`	Missing library	Verify that tcsmgr contains a correct PYTHONPATH. Modify the path or install the library for this particular version of Python
Cannot log in with the given username and password	Wrong or missing user credentials in tcsmgr.pass	Create a new user entry in tcsmgr.pass
Newly installed applications not starting when using start command	An application error	Check the log file using journalctl
Status panel showing: No status received in last <N> seconds	• tcsmgrd not running on the TCS machine • Network connection lost	Log into TCS machine as root and start tcsmgrd service.