



TAUBER TrueSight REST-API Client *II*

User Guide

Version 1.8.50

August 2023



Contacting TAUBER Business Solutions Software

You can access TAUBER Business Solutions Software website at <http://www.tabuso.com> where you can also obtain information about the company, its products and services.

Address

TAUBER Business Solutions Software
Rheinstrasse 35
53332 Bornheim
Germany

Phone: +49 2222 991869

Mail: info@tabuso.com

Web: www.tabuso.com

© Copyright 2008 till 2022 TAUBER Business Solutions UG as an unpublished work. All rights reserved

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc, All other trademarks belong to their respective companies

TAUBER Business Solutions UG considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User Agreement for product and the proprietary and restriction rights included in this document.

Index

Index	3
Version History.....	6
Introduction	9
Introduction	10
Supported OS platforms	10
Supported BMC TrueSight Operations Manager versions	10
Supported BMC Helix Monitor versions	10
Supported Remedy SSO for TrueSight Versions	11
Functions supported by product and version	12
Prerequisites	15
Installation	18
Applying a license	18
Using TRAPIC.....	19
Introduction	19
Authentication	19
Connection data.....	19
Function	19
Output.....	19
Debug.....	20
Connection data.....	20
Cockpit Mode (new)	21
Functions in detail.....	22
getConnectionstatus (TSPS)	22
getauthprofiles (TSPS / TrueSight & Helix Monitor)	22
getdashboardslist (TSPS / TrueSight)	22
getdevicelist (TSPS / TrueSight & Helix Monitor).....	23
getattributelist (TSPS / TrueSight)	23
getmontypes (TSPS / TrueSight)	24
getinstances (TSPS / TrueSight)	24
checkuserauth (TSPS / TrueSight)	24
getpolicylist (TSPS / TrueSight & Helix Monitor).....	25
getperfdata (TSPS / TrueSight)	25
getperfdata2 (TSPS / TrueSight)	26

geteventcounts (TSPS / TrueSight)	27
getdevicecounts (TSPS / TrueSight)	27
getislist (TSPS / TrueSight)	27
getagentlist (TSPS / TrueSight & Helix Monitor)	28
getavmagentlist (TSPS / TrueSight)	28
getapplicationlist (TSPS / TrueSight)	29
getcelllist (TSPS / TrueSight)	29
getcomponentstatus (TSPS / TrueSight)	30
getservicelist (TSPS / TrueSight)	30
getpackages (TSPS / TrueSight)	30
getallserviceinprofile (TSPS / TrueSight)	31
getalldashboardsinprofile (TSPS / TrueSight)	32
getalldevicesinprofile (TSPS / TrueSight)	32
getallapplicationsinprofile (TSPS / TrueSight)	32
getallusergroupsinprofile (TSPS / TrueSight)	33
getalldevicegroupsinprofile (TSPS / TrueSight)	33
getallrolesinprofile (TSPS / TrueSight)	33
getuserlist (RSSO)	34
getrolelist (TSPS / TrueSight)	34
getusergrouplist (TSPS / TrueSight)	35
getdevicegrouplist (TSPS / TrueSight)	35
geteventgrouplist (TSPS / TrueSight)	35
deploypackage (TSPS / TrueSight)	36
searchcis (TSPS / TrueSight)	37
searchdevices (TSPS / TrueSight)	37
Using TRAPIC II wrapper	38
Introduction	38
trgetdevicedata (TSPS / TrueSight)	38
trdevicecompare (TSPS / TrueSight)	40
traddservicetoprofile & trremoveservicefromprofile (TSPS / TrueSight)	41
tradddashboardtoprofile & trremovedashboardfromprofile (TSPS / TrueSight)	42
traddapplicationtoprofile & trremoveapplicationfromprofile (TSPS / TrueSight)	43
tradddevicetoprofile & trremovedevicefromprofile (TSPS / TrueSight)	44
tradddevicegrouptoprofile & trremovedevicegroupfromprofile (TSPS / TrueSight)	45

tradeventgrouptoprofile & trremoveeventgroupfromprofile (TSPS / TrueSight).....	46
trcreateauthprofile & trdeleteauthprofile (TSPS / TrueSight)	47
tradddevicetoapplication & trremovedevicefromapplication (TSPS / TrueSight)	49
trcopyauthprofile (TSPS / TrueSight)	50
trdeletemanageddevice (TSPS / TrueSight)	51
trdeploypackage (TSPS / TrueSight).....	52
trcreateapplication & trdeleteapplication (TSPS / TrueSight)	53
trrestartagent (TSPS / TrueSight).....	54
trgetagentsetup (TSPS / TrueSight)	56
trsetagentsetup (TSPS / TrueSight).....	57
trdeleteagentsetup (TSPS / TrueSight)	58
trgetlicensedata (TSPS / TrueSight)	59
trexportpolicy (TSPS / TrueSight).....	60
trimportpolicy (TSPS / TrueSight)	61
trupdatepolicy (TSPS / TrueSight).....	62
trmodifypolicy (TSPS / TrueSight)	63
trdeletepolicy (TSPS / TrueSight)	64
traudit (TSPS / TrueSight)	65
trexportdashboards (TSPS / TrueSight).....	66
trimportdashboards.exe (TSPS / TrueSight)	67
Trouble shooting	68
Known problems	69

Version History

IMPORTANT

Version II of TRAPIC requires an update of your TRAPIC license file! Licenses for older versions WILL NOT WORK with trapic II !!!! Please contact TAUBER Business Solutions Support to get an updated license file!

V1.0.00

- Supporting BMC Helix Monitor
- Supporting TrueSight 11.3.03
- Added wrapper tradddashboardtoprofile.exe and trremovedashboardfromprofile.exe supporting
- Added function getdashboardlist and getalldashboardsinprofile to trapic II
- Added function gettspsversion to trapic II

V1.0.20

- Bugfixes

V1.1.00

- Added trcopyauthprofile.exe to copy Authorization Profiles within or between TrueSight Systems
- Added SHARED DASHBOARDS to FACL (=Full Access List) in trcreateauthprofile.exe
- Added multiple usergroups and roles in trcreateauthprofile.exe
- Added function getalladminrightsinprofile to trapic
- Fixed a bug in trdeletemanageddevice.exe to delete Managed Devices that exist at multiple TSIMs

NOTE:

The “-p” = Agent Port Option has been changed to “-po” because it was the same as the password option!

- Added option “-setls” to trdeletemanageddevice.exe to reduce REST-API traffic

V1.2.02

- Added function getallnodesinapplication to TRAPIC
- A lot of Bugfixes

V1.3.20

- Added trdeploypackage
- Added tradddevicetoapplication and trremovedevicefromapplication
- Added TrueSight 11.3.04 as supported version
- A lot of bugfixes

V1.4.20

- Added trexportpolicy.exe, trimportpolicy.exe and trdeletepolicy
- Added trapic.conf property for automatic data directory cleanup
- Minor Bugfixes

V1.5.10

- Added trgetagentpackages.exe
- Added trgetagentsetup.exe (Monitoring Studio X require!!!)
- Minor Bugfixes

V1.5.10

- Added trsetagentsetup.exe (Monitoring Studio X require!!!)
- Minor Bugfixes

V1.7.00

- Added TRAPICII functions searchcis and searchdevices
- Added the following TRAPICII wrappers
trcreateapplication.exe, trdeleteapplication.exe trdeleteagentsetup.exe,
trgetlicensedata.exe, trimportpolicy.exe, trexportpolicy.exe,
trmodifypolicy.exe, trupdatepolicy.exe, trdeletepolicy.exe
- Bugfixes

V1.8.22

- Enhancing trdeploypackage.exe
- Adding trexportdashboards.exe & trimportdashboards.exe
- Bugfixes

V1.8.30

- Added trrestartagent.exe
- Bugfixes

V1.8.31

- Adding jq.exe
- Bugfixes

V1.8.40

- Support for 11.3.06
- Smaller Bugfixes & Enhancements

V1.8.50

- Support for 11.3.07
- Smaller Bugfixes & Enhancements

Introduction

This document describes in a fast and simple way the installation, configuration and use of the **TAUBER** TrueSight REST-API Client II (=TRAPIC II).

The product is used to gather and set data using the REST-API build into the BMCs TrueSight Presentation Server which is part of the BMC TrueSight Operations Manager and BMC Helix Monitor solution.

TRAPIC II can also be used to automate processes within TrueSight like adding a Business Service to an Authorization Profile.

For more information about BMC TrueSight Operations Management and BMC Helix Monitor please use www.bmc.com or contact your local BMC office or reseller.

Introduction

Supported OS platforms

This Table shows all supported OS platforms for TRAPIC II.

Operating System	Supported?	Runnig?
Windows 7 (32 & 64 Bit)	Yes	Yes
Windows 10 (32 & 64 Bit)	Yes	Yes
Windows 2008 (32 & 64 Bit)	Yes	Yes
Windows 2012 (32 & 64 Bit)	Yes	Yes
Windows 2016 (32 & 64 Bit)	Yes	Yes
Windows 2019 (32 & 64 Bit)	Yes	Yes

Supported BMC TrueSight Operations Manager versions

The following table shows the supported TrueSight Operations Manager versions:

TrueSight Operations Manager Versions	Supported?	Runnig?
10.7.00 (FixPack 1 bis 3)	Yes	Yes
11.0.00 (FixPack 1 bis 2)	Yes	Yes
11.3.01	Yes	Yes
11.3.02	Yes	Yes
11.3.03	Yes	Yes
11.3.04	Yes	Yes
11.3.05	Yes	Yes
11.3.06	Yes	Yes
11.3.07	Yes	Yes

Supported BMC Helix Monitor versions

The following table shows the supported BMC Helix Monitor versions:

BMC Helix Monitor Versions	Supported?	Running?
1.0.00 (BETA)	No	Partially

Supported Remedy SSO for TrueSight Versions

The following table shows the supported RSSO versions.

Remedy Single Sign On Versions	Supported?	Runnig?
11.0.00 (FixPack 1 bis 2)	Yes	Yes
11.3.01	Yes	Yes
11.3.02	Yes	Yes

Functions supported by product and version

TRAPIC function or wrapper	Remark	Supported Versions			
		TrueSight 10.7.xx	TrueSight 11.0.xx	TrueSight 11.3.xx	Helix 1.0.00 (BETA)
-f getconnectionstatus		Yes	Yes	Yes	Yes
-f getdevicecounts		Yes	Yes	Yes	Yes
-f checklicense		Yes	Yes	Yes	Yes
-f gettspsversion		Yes	Yes	Yes	Yes
-f getdevicelist		Yes	Yes	Yes	No
-f getauthprofiles		Yes	Yes	Yes	Yes
-f getattributelist		Yes	Yes	Yes	No
-f getmontypes		Yes	Yes	Yes	No
-f getinstances		Yes	Yes	Yes	No
-f checkuserauth		Yes	Yes	Yes	No
-f getpolicylist		Yes	Yes	Yes	Yes
-f getperfdata		Yes	Yes	Yes	No
-f getperfdata2		Yes	Yes	Yes	No
-f geteventcounts		Yes	Yes	Yes	Yes
-f getdevicecounts		Yes	Yes	Yes	Yes
-f getislist		Yes	Yes	Yes	No
-f getcelllist		Yes	Yes	Yes	No
-f getagentlist		Yes	Yes	Yes	Yes
-f getapplicationlist		Yes	Yes	Yes	No
-f getrolelist		Yes	Yes	Yes	No
-f getdashboardlist		Yes	Yes	Yes	Yes
-f getusergrouplist		Yes	Yes	Yes	No
-f getdevicegrouplist		Yes	Yes	Yes	No
-f geteventgrouplist		Yes	Yes	Yes	No
-f getcomponentstatus		Yes	Yes	Yes	No
-f getservicelist		Yes	Yes	Yes	No
-f getpackages		Yes	Yes	Yes	No
-f deploypackage		Yes	Yes	Yes	No
-f getalldevicesinprofile		Yes	Yes	Yes	No
-f getallservicesinprofile		Yes	Yes	Yes	No

-f getallapplicationsinprofile		Yes	Yes	Yes	No
-f getallusergroupsinprofile		Yes	Yes	Yes	No
-f getalldevicegroupsinprofile		Yes	Yes	Yes	No
-f getalleventgroupsinprofile		Yes	Yes	Yes	No
-f getallrolesinprofile		Yes	Yes	Yes	No
-f getalldashboardsinprofile		Yes	Yes	Yes	Yes
-f getalladminrightsinprofile		Yes	Yes	Yes	Yes
-f searchcis	> 11.3.04	No	No	Yes	No
-f searchdevices		Yes	Yes	Yes	No
traddapplicationtoprofile.exe		Yes	Yes	Yes	No
tradddashboardtoprofile.exe		Yes	Yes	Yes	No
tradddevicegrouptoprofile.exe		Yes	Yes	Yes	No
traddevicetopprofile.exe		Yes	Yes	Yes	No
tradeventgrouptoprofile.exe		Yes	Yes	Yes	No
traddservicetoprofile.exe		Yes	Yes	Yes	No
tradddevicetoapplication.exe		Yes	Yes	Yes	No
traudit.exe		Yes	Yes	Yes	No
trcopyauthprofile.exe		Yes	Yes	Yes	No
trcreateauthprofile.exe		Yes	Yes	Yes	No
trdeleteauthprofile.exe		Yes	Yes	Yes	No
trdeploypackage.exe		Yes	Yes	Yes	No
trdevicecompare.exe		Yes	Yes	Yes	No
trgetedeviceid.exe		Yes	Yes	Yes	No
trremoveapplicationfromprofile.exe		Yes	Yes	Yes	No
trremovedashboardfromprofile.exe		Yes	Yes	Yes	No
trremovedevicefromprofile.exe		Yes	Yes	Yes	No
trremovedevicegroupfromprofile.exe		Yes	Yes	Yes	No
trremoveeventgroupfromprofile.exe		Yes	Yes	Yes	No
trremovedevicefromapplication.exe		Yes	Yes	Yes	No
trremoveservicefromprofile.exe		Yes	Yes	Yes	No
trcreateapplication (BETA)		Yes	Yes	Yes	No
trdeleteapplication (BETA)		Yes	Yes	Yes	No
trdeletemanageddevice		Yes	Yes	Yes	No
trrestartagent	Requires HW Sentry TSPS Comp.	No	No	Yes	No

trgetagentsetup	Requires HW Sentry TSPS Comp.	No	No	Yes	No
trsetagentsetup	Requires HW Sentry TSPS Comp.	No	No	Yes	No
trdeleteagentsetup	Requires HW Sentry TSPS Comp.	No	No	Yes	No
trgetlicensedata		No	No	Yes	No
trimportpolicy		No	No	Yes	No
trexportpolicy		No	No	Yes	No
trmodifypolicy		No	No	Yes	No
trupdatepolicy		No	No	Yes	No
trdeletepolicy		No	No	Yes	No
trexportpolicy		No	No	Yes	No
trimportpolicy		No	No	Yes	No

Prerequisites

TRAPIC II is a Windows program running on 32 and 64 Bit Windows systems. To use the command line tool make sure you have a user configured that has the privileges to connect to the TrueSight Presentation Server or BMC Helix Monitor.

You should configure a new user, e.g. apiuser and to make sure you can access all parts of the Presentation Server you should add the user to the “Administrators” group.

If you just need access to a separated part of the TSPS you can create a new Role and Authorization profile and reduce the privileges to the required part of TSPS.

IMPORTANT

TRAPIC II is a commercial product that has some functions that can be used free of charge. Below you find a list of free and commercial version functions.

Free Functions	Commercial Version Functions
getconnectionstatus	getperfdata
getdevicelist	getperfdata2
getauthprofiles	geteventcounts
getattributelist	getislist
getmontypes	getagentlist
getinstances	getavmagentlist
checkuserauth	getapplicationlist
getpolicylist	getcomponentstatus
gettspsversion	getservicelist
	getdashboardlist
	getpackages
	getalldevicesinprofile
	getallservicesinprofile
	getallapplicationsinprofile

	getallddevicegroupsinprofile
	getallusergroupsinprofile
	getalleventgroupsinprofile
	getallrolesinprofile
	getalldashboardsinprofile
	getalladminrightsinprofile
	getrolelist
	getusergrouplist
	geteventgrouplist
	getdevicegrouplist
	getuserlist
	searchcis
	searchdevices

TRAPIC II Wrappers
trcopyauthprofile & traudit
trcreateapplication & trdeleteapplication
trcreateauthprofile & trdeleteauthprofile
traddservicetoprofile & trremoveservicefromprofile
traddapplicationtoprofile & trremoveapplicationfromprofile
tradddevicetoapplication & trremovedevicefromapplication
tradddevicetoprofile & trremovedevicefromprofile
tradddevicegrouptoprofile & trremovedevicefromprofile
tradddashboardtoprofile & trremovedashboardfromprofile
traddeventgroupfromprofile & trremoveeventgroupfromprofile
trdeletemanageddevice & trgetdevicedata & trdevicecompare
trdeploypackage & trgetlicensedata
trgetagentsetup & trsetagentsetup & trdeleteagentsetup
trimportpolicy & trexportpolicy & trupdatepolicy & trmodifypolicy & trdeletepolicy

NOTE:

TRAPIC II wrappers are additional command line tools that use trapic.exe function calls to implement more complex processes using the TrueSight

Presentation Server or BMC Helix Monitor REST-API like adding an object to an Authorization Profile.

All TRAPIC II wrappers are commercial tools and need a valid license!

To use the functions marked as commercial contact support@tabuso.com and make sure that you place the license.txt into the same directory all .exe files are located in.

To use TRAPIC II you need a copy of the command line tool curl.exe. The latest release of TRAPIC II contains a working copy of curl which is by default used. It is located in the TRAPIC II subdirectory called "curl"!

Installation

To install TRAPIC II download the product installation package from the TAUBER Business Solutions Ug website and unzip it on your Windows system. Make sure you have execute privileges in this directory.

Next to this you need to be able to connect to the HTTPS port of your TrueSight Presentation Server or BMC Helix Monitor which by default is 443 if you run TrueSight on Windows and 8043 on Linux.

Execute trapic.exe without any arguments to check your license.

```
D:\trapic>trapic.exe
(T)ruesight (R)EST (API) (C)lient V3.0.00
(c) TAUBER Business Solutions Ug
written by Timo Schmidt / http://www.tabuso.com/
No license.txt found ... using trial version!

trapic.exe [options] [function]
```

Pic. 1: trapic.exe without license!

If you have no valid license TRAPIC tells you by showing “No license ...”. If you have no license you can use the free functions.

Applying a license

If you receive a commercial license you get a text file called license.txt. Copy the file to the same directory you have trapic.exe in and execute “trapic.exe”. TRAPIC II will tell you the license details if it accepts the license.

```
(T)ruesight (R)est (API) (C)lient U2.0.00
(c) TAUBER Business Solutions Ug
written by Timo Schmidt / http://www.tabuso.com,
Customer name: TAUBER / Max # session: 0005!
Expiration data 01.10.2018!
```

Pic. 2: TRAPIC II with valid license!

Using TRAPIC

Introduction

The TRAPIC II command line tool arguments consist of five different areas some mandatory and some optional based on the connection type and target system (see examples below the list):

Authentication

- u [Username] The name of the user to use for the authentication
- p [Password] The password of the user
- te [Tenant] The tenant you want to authenticate at

Note: The uber tenant is

BmcRealm	<= 10.7.xx
*	> 10.7.xx

- ts [Server]:[Port] The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
- tt [truesight| |helix] The type of target system you are connecting to. By default the type is truesight. To connect to Helix Monitor you have to add "-tt "helix"
- rs [Server]:[Port] The Rest-API of the RSSO Sever in [Hostname]:[Port] format
- rv [Release] The RSSO version 11.0 or 11.3

Connection data

- ss [Filename] Store connection data in file
- cs [Filename] Connect to system stored in file

Function

- f [function]

Output

- kof This will keep the output file instead of deleting it after the execution. The filename is called trapic.exe.out
- csv This will generate a comma separated list output
- tbl This will generate an output in table format

Debug

-debug Activate the debug output

This is an example to generate a list of devices from a TrueSight 11.3 System:

Example 1

```
trapic.exe -u "apiuser" -p "XYZXYZXYZ" -te * -ts "ts-server:443" -f getdevicelist
```

Output

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f getdevicelist
(T)ruessight (R)est (API) (C)lient V2.0.00
(c) TAUBER Business Solutions UG
written by Timo Schmidt / http://www.tabuso.com/
Customer name: TAUBER / Max # session: none!
Expiration data 01.01.2019!
```

```
Device # [1] / Device ID = [3] / Device DNS name = [ptf-sles-52.truesight.tabuso.com] / Device Display name = [ptf-sles-52.truesight.tabuso.com] / Server ID = [2]
Device # [2] / Device ID = [7] / Device DNS name = [ptf-sles-51.truesight.tabuso.com] / Device Display name = [ptf-sles-51.truesight.tabuso.com] / Server ID = [2]
Device # [3] / Device ID = [1] / Device DNS name = [ptf-sles-54.truesight.tabuso.com] / Device Display name = [ptf-sles-54.truesight.tabuso.com] / Server ID = [2]
Device # [4] / Device ID = [10] / Device DNS name = [ptf-win-7.truesight.tabuso.com] / Device Display name = [ptf-win-7.truesight.tabuso.com] / Server ID = [2]
Device # [5] / Device ID = [4] / Device DNS name = [ptf-sles-53.truesight.tabuso.com] / Device Display name = [ptf-sles-53.truesight.tabuso.com] / Server ID = [2]
Device # [6] / Device ID = [6] / Device DNS name = [ptf-ws08-ora05.truesight.tabuso.com] / Device Display name = [ptf-ws08-ora05.truesight.tabuso.com] / Server ID = [2]
Device # [7] / Device ID = [9] / Device DNS name = [ptf-sles-56.truesight.tabuso.com] / Device Display name = [ptf-sles-56.truesight.tabuso.com] / Server ID = [2]
Device # [8] / Device ID = [5] / Device DNS name = [ptf-sles-55.truesight.tabuso.com] / Device Display name = [ptf-sles-55.truesight.tabuso.com] / Server ID = [2]
Device # [9] / Device ID = [2] / Device DNS name = [ptf-sles-50.truesight.tabuso.com] / Device Display name = [ptf-sles-50.truesight.tabuso.com] / Server ID = [2]
Device # [10] / Device ID = [8] / Device DNS name = [ptf-sles-57.truesight.tabuso.com] / Device Display name = [ptf-sles-57.truesight.tabuso.com] / Server ID = [2]
```

Total number devices = [10]

Example 2

Connect to a Helix Monitor and get the list of Devices.

```
trapic.exe -u "apiuser" -p "XYZXYZXYZ" -te * -ts -tt "helix" "ts-server:443" -f getdevicelist
```

Connection data

You can store and reuse the connection data (Username, Password, Tenant, TSPS Server + Port). The password is encrypted in a simple and proprietary way! The file itself is a text file with "[Property]=[Value]" format. Despite the encrypted password you can change all other values directly in the file.

To store the connection data, execute the command as always to get some data from your TSPS and add the "-ss" (= store system config) option with a folder + file combination, e.g.

```
trapic.exe -u admin -p XYZXYZ -te * -ts "ptf-sles-51:8043" -ss "config\TS11.cfg" -f getislist
```

This will store the connection data in the folder “config” and the file TS11.cfg! To use it with your next command use

```
tragic.exe -cs "config\TS11.cfg" -f getislist
```

Connection file handling options:

-ss [Filename] Stores the connection data in file.

-cs [Filename] Uses the connection data from a file.

Cockpit Mode (new)

The Cockpit mode can be used with some first functions to constantly monitor and observe the status of the TrueSight System. The following functions support Cockpit mode:

getcomponentstatus / getdevicecounts / geteventcounts

To run the function in cockpit mode, add the argument “-cockpit” to the function call. TRAPIC II will start running and displaying the data in table mode and count down – by default five seconds – to refresh the display.

```
Select Administrator: Command Prompt - tragic.pl -cs configs\TrueSight113.cfg -f getcomponentstatus -cpit
TrueSight (R)EST (API) (C)lient V3.2.xx / getdevicestatus / Cockpit mode
```

#	Component ID	Component DNS Name	Component Type	Component Description	Connection Status
1	8	ptf-sles-56.truesight.tabuso.com	APM	App Visibility Portal	DISCONNECTED
Sub Component Details for 1.1 Collectors:					
#	Sub Component Name	Sub Component Total #	Sub Component Running #	Sub Component NOT Running #	
1.1	Collectors	1	0	1	
1.2	Proxies	0	0	0	
1.3	Agents	0	0	0	
1.4	Synthetic TEA Agents	1	0	1	
2	12	ptf-sles-57.truesight.tabuso.com	ITDA	IT Data Analytics Server	INITIALIZING_ERROR
Sub Component Details for 2.1 Collection Station:					
#	Sub Component Name	Sub Component Total #	Sub Component Running #	Sub Component NOT Running #	
2.1	Collection Station	1	0	1	
2.2	Indexer	1	0	1	
2.3	Search	1	0	1	
2.4	Console	1	0	1	
2.5	Configuration DataBase1	1	0	1	
3	17	tsim113lb.truesight.tabuso.com	TSIM	Infrastructure Management Server	CONNECTED

```
### Note: Press q on your keyboard while the status is updated or use CTRL + C to quit!
Updating in ... 5
```

Pic. 3: getcomponentstatus in Cockpit mode

You can change the refresh delay by adding the argument “-ut” (=update time) and a value in seconds. Please keep in mind that every X seconds a REST-API request is send so a minimal refresh time of one second will have an impact on your presentation server. Using the default of five should have a really small impact!

Example

```
tragic.exe -cs configs\TrueSight113.cfg -f getcomponentstatus -cockpit -ut 10
```

Functions in detail

The following functions can be used after the “-f” argument

getconnectionstatus (TSPS)

Arguments:

none

The getconnectionstatus function shows the status of the TSPS REST-API connection.

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getconnectionstatus
```

Example Output:

```
Connection status OK
```

getauthprofiles (TSPS / TrueSight & Helix Monitor)

Arguments: none

The getauthprofiles function shows all Authorization Profiles defined in the Presentation Server.

Example Command:

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f getconnectionstatus
```

Example Output:

```
# 1 / Authprofile name = [TDS_Unit_ReadOnly] / Authprofile id = [ebccccdd-fc96-4fe4-b1f0-6e89421e8392] / Authprofile owner = [admin] / Authprofile tenant = [bmcrealm]
# 2 / Authprofile name = [API-Only User] / Authprofile id = [deb70416-63c0-4e65-b265-c6dad241b584] / Authprofile owner = [admin] / Authprofile tenant = [BmcRealm]
# 3 / Authprofile name = [Capacity Administration] / Authprofile id = [1c37e280-38d2-4d72-89b8-dc73238e696e] / Authprofile owner = [admin] / Authprofile tenant = [BmcRealm]
# 4 / Authprofile name = [Executive] / Authprofile id = [91d86403-184f-47e0-9bd9-997b03017a7d] / Authprofile owner = [admin] / Authprofile tenant = [BmcRealm]
...
```

getdashboardslist (TSPS / TrueSight)

Arguments: none

The getdashboardlist function shows all Shared Dashboards defined in the Presentation Server.

Example Command:

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f getdashboardlist
```

Example Output:

```
# 1 / Id = [1] / Dashboard name = [ITDA Test Dashboard] / Dashboard owner = [admin]
# 2 / Id = [2] / Dashboard name = [TEST Dashboard] / Dashboard owner = [admin]...
```

getdevicelist (TSPS / TrueSight & Helix Monitor)

Arguments: none

The getdevicelist function requests all devices from the TSPS and shows you

- Device ID of the device
- Device DNS Name of the device
- Device Display Name of the device
- Server ID of the TSIM the device is connected to

Example Command:

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f getdevicelist
```

Example Output:

```
Device # [1] / Device ID = [3] / Device DNS name = [ptf-sles-52.truesight.tabuso.com] / Device Display name = [ptf-sles-52.truesight.tabuso.com] / Server ID = [2]
...
Device # [2] / Device ID = [7] / Device DNS name = [ptf-sles-51.truesight.tabuso.com] / Device Display name = [ptf-sles-51.truesight.tabuso.com] / Server ID = [2]
Total number devices = [10]
```

getattributelist (TSPS / TrueSight)

Arguments:

- | | | |
|----------------------|----------|---|
| -did [device id] | Required | Device ID of the device you want to query all attributes for. |
| -str [search string] | Optional | Searchstring to search the attributes for |

The getattributelist function requests all attributes for a device. You have to pass the device ID (see getdevicelist command) to the function. An optional argument is a search string. The function shows you

- The Device ID
- The Server ID
- The Monitor Type ID of the attribute
- The Monitor Instance ID of the attribute
- The Attribute name of the attribute
- The Attribute ID of the attribute
- The Attribute Key (used e.g. to get performance data with getperfdata)

Example Command:

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f getattributelist -did 10 -str "Cache Faults"
```

Example Output:

```
# 1 / Device ID = [5] / Server ID = [2] / Monitor Type ID = [501016] / Monitor Instance ID = [398] / Attribute name = [CPU::Linux OS/CPU:Utilization] / Attribute ID = [501016507] / Attribute Key = [5+2+501016+398+501016507]
# 2 / Device ID = [5] / Server ID = [2] / Monitor Type ID = [501016] / Monitor Instance ID = [398] / Attribute name = [CPU::Linux OS/CPU:Utilization In System Mode] / Attribute ID = [501016508] / Attribute Key = [5+2+501016+398+501016508]
# 3 / Device ID = [5] / Server ID = [2] / Monitor Type ID = [501016] / Monitor Instance ID = [398] / Attribute name = [CPU::Linux OS/CPU:Utilization In User Mode] / Attribute ID = [501016509] / Attribute Key = [5+2+501016+398+501016509]
```

getmontypes (TSPS / TrueSight)

Arguments:

-cat [all|instances|group] Optional Category = all | instances | group

The getmontypes function shows you all monitor types and for each the following information

- The Monitor name
- The Monitor category

Example Command:

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f getmontypes -cat group
```

Example Output:

```
# 1 / Monitor type # [1] / Monitor name = [Compag Correctable Memory Error (Deprecated)] / Monitor unique name = [cpqHeCorrMe] / Monitor category = [instance]
# 2 / Monitor type # [2] / Monitor name = [IIS Active Server Pages] / Monitor unique name = [IISASPMonInfo] / Monitor category = [instance]
```

getinstances (TSPS / TrueSight)

Arguments:

-did Required ID of the device monitor instance data should be collected from.

The getinstances function shows monitor type and instance data for the device selected and for each instance the following information

- The Monitor unique name
- The PATROL Application class
- The Monitor Instance name
- Is it Marked for delete?
- The Server Id
- The Monitor Type Id
- The Monitor Instance Id

Example Command:

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f getinstances -did 50
```

Example Output:

```
# 1 / Monitor unique name = [_PATROL__NUK_KERNEL] / PATROL App class = [NUK_KERNEL] / Monitor Instance name = [Kernel] / MFD? = [false] / Server id = [8] / Monitor type id = [501016] / Moitor instance id = [394]
# 2 / Monitor unique name = [_PATROL__NUK_NETWORK_CONTAINER] / PATROL App class = [NUK_NETWORK_CONTAINER] / Monitor Instance name = [Network] / MFD? = [false] / Server id = [8] / Monitor type id = [501022] / Moitor instance id = [336]
# 3 / Monitor unique name = [_PATROL__NUK_PROCESS] / PATROL App class = [NUK_PROCESS] / Monitor Instance name = [bind] / MFD? = [false] / Server id = [8] / Monitor type id = [501028] / Moitor instance id = [433]
```

checkuserauth (TSPS / TrueSight)

Arguments:

-user [user] Required username to check Groups and Authorization Profiles for

The checkuserauth function helps you to find all Groups and Authorization Profiles a user is in and shows the following information

- The Group name of the group the user is in
- The Authorization Profile the user is in by the group he is in

Example Command:

```
trapic.exe -u admin -p XYZXYZXYZ -te * -ts ptf-sles-50:8043 -f checkuserauth -user admin
```

Example Output:

```
# 1 / Username = [admin] / Group Name = [Capacity_Administration] / Authorization Profile = [Capacity Administration]
# 2 / Username = [admin] / Group Name = [Administrators] / Authorization Profile = [Solution Administrator]
```

getpolicylist (TSPS / TrueSight & Helix Monitor)

Arguments:

-sp -bp -mp	Optional	Type of Policy
		(sp = Staging Policy / bp = Blackout Policy / Mp = Monitoring Policy)

The getpolicylist function requests by default all Policies configured in the TSPS. By using the optional arguments the function just shows the requested type of Policies. For each Policy it returns

- The name of the Policy
- The precedence of the Policy
- The owner of the Policy
- Is it enabled?

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getpolicylist
```

Example Output:

```
# 1 / Policy name [NT_WINDOWS_SVC_RESTART_V_4_9_00] / Policy precedence = [700] / Policy owner = [admin] / Policy enabled? = [true]
# 2 / Policy name [AGT_GRP_TEST_1] / Policy precedence = [900] / Policy owner = [admin] / Policy enabled? = [true]
# 3 / Policy name [NT_KEYNOTE_KM_V0_9_12] / Policy precedence = [700] / Policy owner = [admin] / Policy enabled? = [true]
# 4 / Policy name [LX_NUK_KM_REMOTE_1_2_00_01] / Policy precedence = [702] / Policy owner = [admin] / Policy enabled? = [false]
```

getperfdata (TSPS / TrueSight)

Arguments:

-key [attribute key]	Required	Attribute key
-time [mins]	Required	Timeframe in minutes from now into the past

The getperfdata function requests the raw data from the TSIM database through the TSPS Rest-API. To get the data you have to provide the attribute key which is unique for each attribute and the time frame in minutes you want raw data for. The function shows the following values for each data point:

- A readable date & time string
- An epoch date & time value
- The data point value

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getperfddata -key 15+8+501034+523+501034502 -time 10
```

Example Output:

```
# 1 / Data point date & time = [10.11.2018 13:30:49] / Data point timestamp (epoch) = [1541853049] / Data point value = [5.29]
# 2 / Data point date & time = [10.11.2018 13:31:49] / Data point timestamp (epoch) = [1541853109] / Data point value = [1.56]
# 3 / Data point date & time = [10.11.2018 13:32:49] / Data point timestamp (epoch) = [1541853169] / Data point value = [1.64]
# 4 / Data point date & time = [10.11.2018 13:33:49] / Data point timestamp (epoch) = [1541853229] / Data point value = [1.09]
```

getperfddata2 (TSPS / TrueSight)

Arguments:

-sid [TSIM Server ID]	Required	TSIM Server ID (use getdevicelist to find)
-mt [Monitor Type ID]	Required	Monitor Type ID (use getattributelist to find)
-mi [Monitor Inst. ID]	Required	Monitor Instance ID (use getattributelist to find)
-at [Attribute ID]	Required	Attribute ID (use getattributelist to find)
[-time [mins]]	Optional	Timeframe in minutes from now into the past]

Or

[-st start time] -et [end time]	Start and Endtime in Epoch
-pad	Optional Get data from the PATROL Agent

The getperfddata2 function requests the raw data from the TSIM database through the TSPS Rest-API. The getperfddata2 function is much more flexible and easier to use because you can provide the Ids, grep for a time period of data and get the data from the PATROL Agent himself:

- A readable date & time string
- An epoch date & time value
- The data point value

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts my-monitoring.dyndns-free.com:8043 -f getperfddata2 -sid 8 -mt 501042 -mi 350 -at 501042507 -time 10
```

Example Output:

```
# 1 / Data point date & time = [10.11.2018 13:30:49] / Data point timestamp (epoch) = [1541853049] / Data point value = [5.29]
# 2 / Data point date & time = [10.11.2018 13:31:49] / Data point timestamp (epoch) = [1541853109] / Data point value = [1.56]
# 3 / Data point date & time = [10.11.2018 13:32:49] / Data point timestamp (epoch) = [1541853169] / Data point value = [1.64]
# 4 / Data point date & time = [10.11.2018 13:33:49] / Data point timestamp (epoch) = [1541853229] / Data point value = [1.09]
```

geteventcounts (TSPS / TrueSight)

Arguments:

-cockpit	Optional	Run in cockpit mode
-ut [Time in Seconds]	Optional	Update time

The geteventcounts function requests the event count values for different categories like total events, events by status and severity:

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts my-monitoring.dyndns-free.com:8043 -f geteventcounts
```

Example Output:

```
# Events Status open = [170] / # Events Status acknowledged = [0] / # Events Status assigned = [0] / # Events Status blackout = [0] / # Events Status closed = [120]
# Events Severity Critical = [8] / # Events Severity Major = [0] / # Events Severity Minor = [0] / # Events Severity Warning = [141] / # Events Severity Info = [125] / # Events Severity OK = [16] / # Events
Severity Unknown = [0] Total # Events = [290]
```

getdevicecounts (TSPS / TrueSight)

Arguments:

-cockpit	Optional	Run in cockpit mode
-ut [Time in Seconds]	Optional	Update time

The getdevicecount function requests the device count values for different categories like total devices, devices by status and severity:

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts my-monitoring.dyndns-free.com:8043 -f getdevicecounts
```

Example Output:

```
# Devices Status open = [3] / # Devices Status acknowledged = [0] / # Devices Status assigned = [0] / # Devices Status blackout = [0] / # Devices Status closed = [0]
# Devices Severity Critical = [0] / # Devices Severity Major = [0] / # Devices Severity Minor = [0] / # Devices Severity Warning = [0] / # Devices Severity Info = [3]
Total # Devices = [3]
```

getislist (TSPS / TrueSight)

Arguments: none

The getislist function requests all ISes connected to the TSIMs of this TrueSight Infrastructure and shows the following information for each IS:

- The logical IS name
- The IS hostname
- The IS version
- The IS port
- The IS staging port
- The IS type (staging or normal)
- The IS connection status

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getislist
```

Example Output:

```
# 1 / TSIM ID = [8] / TSIM DNS name = [tsim03lb.truesight.tabuso.com] / IS Name = [ptf-sles-24.truesight.tabuso.com] / IS Hostname = [ptf-sles-24.truesight.tabuso.com] / IS version = [TrueSight Integration Service 10.7 build 321427196] / IS port = [12124] / IS staging port = [3183] / IS type = [normal] / IS Connection Status = [CONNECTED]
# 2 / TSIM ID = [8] / TSIM DNS name = [tsim03lb.truesight.tabuso.com] / IS Name = [ptf-sles-25.truesight.tabuso.com] / IS Hostname = [ptf-sles-25.truesight.tabuso.com] / IS version = [TrueSight Integration Service 10.7 build 321427196] / IS port = [12124] / IS staging port = [3183] / IS type = [normal] / IS Connection Status = [CONNECTED]
# 3 / TSIM ID = [8] / TSIM DNS name = [tsim03lb.truesight.tabuso.com] / IS Name = [TSIM03_IS_01] / IS Hostname = [ptf-sles-17.truesight.tabuso.com] / IS version = [TrueSight Integration Service 10.7 build 241368724] / IS port = [12124] / IS staging port = [3183] / IS type = [normal] / IS Connection Status = [CONNECTED]
# 4 / TSIM ID = [8] / TSIM DNS name = [tsim03lb.truesight.tabuso.com] / IS Name = [TSIM03_IS_02] / IS Hostname = [ptf-sles-16.truesight.tabuso.com] / IS version = [TrueSight Integration Service 10.7 build 241368724] / IS port = [12124] / IS staging port = [3183] / IS type = [normal] / IS Connection Status = [CONNECTED]
# 5 / TSIM ID = [8] / TSIM DNS name = [tsim03lb.truesight.tabuso.com] / IS Name = [TSIM03_SIS_01] / IS Hostname = [ptf-sles-18.truesight.tabuso.com] / IS version = [TrueSight Integration Service 10.7 build 241368724] / IS port = [12124] / IS staging port = [3183] / IS type = [normal] / IS Connection Status = [CONNECTED]
Total # ISes = [5] / Number ISes connected [0] / Number ISes disconnected [5]
```

getagentlist (TSPS / TrueSight & Helix Monitor)

Arguments:

-setls	Optional	Integration Service Hostname + Port
-agtCon	Optional	Show just connected agents
-agtDis	Optional	Show just disconnected agents

The getagentlist function requests the status of the connected PATROL Agents and shows

- The Agent Id
- The Agent name
- The Agent port
- The Agent version
- The Agent Operating System
- The Monitor instance Id
- The Integration Service Name the agent is connected to
- The Integration Service Hostname the agent is connected to
- The Deployment Status

Example Command:

```
tragic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getavmagentlist -pid 4
```

Example Output:

```
...
# 1 / Agent Id = [11] / Agent name [ptf-sles-3.pnet.tabuso.com] / Agent Port [3181] / Agent version = [V9.6.00.1i] / Agent connection state = [Disconnected] / Monitor Instance Id = [10016] / Agent IP Address [127.0.0.2] / Integration Service Name = [ProactiveServer] / Integration Service Hostname = [ptf-ws16-tsim.tabuso.com]
# 2 / Agent Id = [9] / Agent name [ptf-ws16-tsim.tabuso.com] / Agent Port [3181] / Agent version = [V11.3.01i] / Agent connection state = [Connected] / Monitor Instance Id = [10014] / Agent IP Address [192.168.148.182] / Integration Service Name = [ProactiveServer] / Integration Service Hostname = [ptf-ws16-tsim.tabuso.com]
# 3 / Agent Id = [8] / Agent name [ptf-ws16-tsp.tabuso.com] / Agent Port [3181] / Agent version = [V11.3.01i] / Agent connection state = [Connected] / Monitor Instance Id = [10013] / Agent IP Address [192.168.148.181] / Integration Service Name = [ProactiveServer] / Integration Service Hostname = [ptf-ws16-tsim.tabuso.com]
Total # Agents = [3] / Agents connected = [2] / Agents disconnected = [1]
```

getavmagentlist (TSPS / TrueSight)

Arguments:

-pid [PortalID]	Required	AVM Portal ID (see getcomponentstatus)
-----------------	----------	--

The getavmagentlist function requests the status of the connected AVM agents and provides the following information

- The logical Agent name

- The Agent address / hostname
- The Agent platform
- The Agent connection status

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getavmagentlist -pid 4
```

Example Output:

```
# 1 / AVM Agent name = [TEA_AGENT_01] / AVM Agent address = [ptf-win-5] / AVM Agent platform = [Synthetic] / AVM Agent status = [ONLINE]
# 2 / AVM Agent name = [TEA_AGENT_02] / AVM Agent address = [ptf-win-22] / AVM Agent platform = [Synthetic] / AVM Agent status = [ONLINE]
```

getapplicationlist (TSPS / TrueSight)

Arguments:

none

The getapplicationlist function shows the following information for all applications in the TSPS:

- The Application Id
- The Application display name
- The Application severity
- The time since when the severity occurred in Epoch
- The time since when the severity occurred in Date & Time

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getapplicationlist
```

Example Output:

```
# 1 / Application Id = [36] / Application name = [Timo] / Application severity = [OK] / Application severity since = [1542707519763] / Application severity since date & time = [Jun 15 50856 04:36:03]
# 2 / Application Id = [19] / Application name = [Katz&Co] / Application severity = [OK] / Application severity since = [1542707519763] / Application severity since date & time = [Jun 15 50856 04:36:03]
# 3 / Application Id = [11] / Application name = [TrueSight Application] / Application severity = [WARNING] / Application severity since = [1540059975423] / Application severity since date & time = [Jul 23 50772 07:37:03]
# 4 / Application Id = [18] / Application name = [TAUBER Business Solutions] / Application severity = [OK] / Application severity since = [1542707519763] / Application severity since date & time = [Jun 15 50856 04:36:03]
```

getcelllist (TSPS / TrueSight)

Arguments:

none

The getcelllist function shows the following information for all cells registered at infrastructure cells in the TSPS:

- The Cell name
- The Cell host
- The Cell port
- The secondary Cell host
- The secondary Cell port

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getcelllist
```

Example Output:

```
# 1.1 / TSIM name = [tsim03lb.truesight.tabuso.com] / Cell Name = [admin] / Cell Host = [ptf-sles-24] / Cell Port = [1827] / Cell Secondary Host = [ptf-sles-25.truesight.tabuso.com] / Cell Secondary Port = [1827]
```

```
Total # TSIMs = [1] / Total # Cells = [1]
```

getcomponentstatus (TSPS / TrueSight)

Arguments:

-cockpit	Optional	Run in cockpit mode
-ut [Time in Seconds]	Optional	Update time

The getcomponentstatus function requests the status of all components configured in TSPS. The function provides the following information of each component:

- Component ID
- Component DNS name
- Component type
- Component description
- Connection status

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getcomponentstatus
```

Example Output:

```
# 1 / Component ID [4] / Component DNS name = [ptf-sles-15.truesight.tabuso.com] / Component type = [APM] / Component description = [App Visibility Portal] / Connection status = [CONNECTED]
# 2 / Component ID [8] / Component DNS name = [tsim03lb.truesight.tabuso.com] / Component type = [TSIM] / Component description = [Infrastructure Management Server] / Connection status = [CONNECTED]
```

```
# 4.1 / name = [Linux] / Solution version = [1.2.00.01]
```

getservicelist (TSPS / TrueSight)

Arguments:

none

The getservicelist function shows the following information for all services in the TSPS:

- The Service name
- The Service display name
- The Service status

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getservicelist
```

Example Output:

```
# 1 / Service name = [TSIM Service] / Service Display Name = [TSIM Service] / Service status = [UNAVAILABLE]
# 2 / Service name = [TSPS Service] / Service Display Name = [TSPS Service] / Service status = [UNAVAILABLE]
```

getpackages (TSPS / TrueSight)

Arguments:

-sol	Optional	Show all solutions included in a Deployable Package
------	----------	---

The `getpackages` function shows the following information for all Deployable Packages defined and optional also the solutions included. For each package it shows

- The package name
- The OS name
- The platform
- The template name
- The installation directory

For each solution it shows

- Solution name
- Solution version

Example Command:

```
trapic.exe -u apiuser -p XYZXYZYZ -te BmcRealm -ts ptf-sles-9:8043 -f getpackages -sol
```

Example Output:

```
# 1 / Package name [PTF_DIR_MON_KM] / OS Name = [Windows] / Platform Name = [x64] / Template Name = [PTF_DIR_MON_KM] / Install dir = [C:\Program Files\BMC Software\PAgent]
# 1.1 / name = [Directory Monitor] / Solution version = [2.1.22]
# 2 / Package name [IKEA_F5_V1_1_02] / OS Name = [Windows] / Platform Name = [x64] / Template Name = [IKEA_F5_V1_1_02] / Install dir = [C:\Program Files (x86)\BMC Software]
# 2.1 / name = [IKEA F5 Monitor] / Solution version = [1.1.02]
# 3 / Package name [PATROL_Classic_Console] / OS Name = [Windows] / Platform Name = [x64] / Template Name = [PATROL_Classic_Console] / Install dir = [C:\Program Files\BMC Software\PAgent]
# 3.1 / name = [PATROL Console] / Solution version = [3.6.00]
# 4 / Package name [PATROL_Agent_and_NUK] / OS Name = [Linux] / Platform Name = [x64] / Template Name = [PATROL_Agent_and_NUK] / Install dir = [/opt/bmc/PAgent]
```

getallserviceinprofile (TSPS / TrueSight)

Arguments:

-an Required [Authprofile name]

The `getallservicesinprpfile` function shows the following information for all Services in the Authorization profile:

- The Service name
- The Service ID

Example Command:

```
trapic.exe -u apiuser -p XYZXYZYZ -te BmcRealm -ts ptf-sles-9:8043 -f getallservicesinprofile -an "Mein Test Profil"
```

Example Output:

```
# 1 / Service name = [Timos Test Service] / Service id = [17+mc.pncell_ptf-sles-54.1ca48dc9.0]
```

getalldashboardsinprofile (TSPS / TrueSight)

Arguments:

-an Required [Authprofile name]

The getalldashboardsinprofile function shows the following information for all Shared Dashboards in the Authorization profile:

- The Dashboard Id
- The Dashboard name
- The Component Id

Example Command:

```
tragic.exe -u apiuser -p XYZXYZYZ -te BmcRealm -ts ptf-sles-9:8043 -f getalldashboardsinprofile -an "Mein Test Profil"
```

Example Output:

```
# 1 / Id = [1] / Dashboard name = [ITDA Test Dashboard] / Component Id = [ptf-ws16-tsp.tabuso.com]
# 2 / Id = [2] / Dashboard name = [TEST Dashboard] / Component Id = [ptf-ws16-tsp.tabuso.com]
```

getalldevicesinprofile (TSPS / TrueSight)

Arguments:

-an Required [Authprofile name]

The getalldevicesinprofile function shows the following information for all Devices in the Authorization profile:

- The Device name
- The Device ID

Example Command:

```
tragic.exe -u apiuser -p XYZXYZYZ -te BmcRealm -ts ptf-sles-9:8043 -f getalldevicesinprofile -an "Mein Test Profil"
```

Example Output:

```
# 1 / Device name = [ptf-sles-55.truesight.tabuso.com] / Device id = [91]
# 2 / Device name = [ptf-sles-56.truesight.tabuso.com] / Device id = [94]
```

getallapplicationsinprofile (TSPS / TrueSight)

Arguments:

-an Required [Authprofile name]

The getallapplicationsinprofile function shows the following information for all Applications in the Authorization profile:

- The Application name
- The Application ID

Example Command:

```
tragic.exe -u apiuser -p XYZXYZYZ -te BmcRealm -ts ptf-sles-9:8043 -f getallapplicationsinprofile -an "Mein Test Profil"
```

Example Output:

1 / Application name = [Timos Test Application] / Application Id = [0+9bbaf693-44d6-401c-98eb-67b299400fb2A]
 # 2 / Application name = [Test Application 2] / Application Id = [0+f8655037-3e50-40f2-9396-0364aa7a78982A]

getallusergroupsinprofile (TSPS / TrueSight)

Arguments:

-an Required [Authprofile name]

The getallusergroupsinprofile function shows the following information for all Usegroups in the Authorization profile:

- The Usergroup name
- The Usergroup Id

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getallusergroupsinprofile -an "Mein Test Profil"
```

Example Output:

```
# 1 / Device Group name = [TEST] / Device Group id = [0,TS5]  
# 2 / Device Group name = [BMC_TrueSightAgent] / Device Group id = [17,3]
```

getalldevicegroupsinprofile (TSPS / TrueSight)

Arguments:

-an Required [Authprofile name]

The getallusergroupsinprofile function shows the following information for all Usegroups in the Authorization profile:

- The Device Group name
- The Device Group Id

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getalldevicegroupsinprofile -an "Mein Test Profil"
```

Example Output:

```
# 1 / Device Group name = [TEST] / Device Group id = [0,TS5]  
# 2 / Device Group name = [BMC_TrueSightAgent] / Device Group id = [17,3]
```

getallrolesinprofile (TSPS / TrueSight)

Arguments:

-an Required [Authprofile name]

-dpd Optional Detailed Permission Data

The getallrolesinprofile function shows the following information for all Roles in the Authorization profile. If you add the "-dpd" Option all Permissions will be listed!

- The Role name
- The Role Id

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getallrolesinprofile -an "Mein Test Profil"
```

Example Output:

```
# 1 / Role name [Test Role] / Role Id = [99c2a35e-5ec5-4fe9-80da-bb0f144cf430]
# 2 / Role name [Read Only] / Role Id = [f636e4dc-0755-4a94-af19-0651d04e3d1c]
```

getuserlist (RSSO)

Arguments: none

The `getuserlist` function connects to Remedy SSO for TrueSight and requests the list of connected users. The following information is provided for all connected users:

- The User Id
- The Creation time
- The Tenant
- The Session timeout

Example Command:

```
trapic.exe -u Admin -p XYZXYZXYZ -rv 11.3 -rs ptf-win-7.truesight.tabuso.com:448 -f getuserlist -debug -kof
```

Example Output:

```
User Id = [App_Visibility_Internal_F2A5A7EC] / Creation time = [Oct 5 2018 12:40:09] / Tenant = [*] / Session timeout = [Oct 6 2018 12:40:09]
User Id = [admin] / Creation time = [Oct 6 2018 08:58:21] / Tenant = [*] / Session timeout = [Oct 7 2018 08:58:21]
User Id [admin] / Creation time = [Oct 6 2018 08:58:21] / Tenant = [*] / Session timeout = [Oct 7 2018 08:58:21]
```

getrolelist (TSPS / TrueSight)

Arguments:

none

The `getrolelist` function shows all roles defined in the Presentation Server with the following information:

- The Role name
- The Role owner
- The Role tenant
- The Role Id

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getrolelist
```

Example Output:

```
# 1 / Role name = [Test Role] / Role owner = [admin] / Role tenant = [*] / Role Id = [99c2a35e-5ec5-4fe9-80da-bb0f144cf430]
# 2 / Role name = [Super Admin] / Role owner = [admin] / Role tenant = [*] / Role Id = [ed390245-5b05-497e-aa4b-0d73e8750499]
# 3 / Role name = [Monitoring Administrator] / Role owner = [admin] / Role tenant = [*] / Role Id = [eefe9657-f768-447a-97d6-f6b4eda66e37]
# 4 / Role name = [Application Administrator] / Role owner = [admin] / Role tenant = [*] / Role Id = [9b7ea76d-2e2c-47a3-8c6c-63313e45f6b9]
# 5 / Role name = [App Visibility Internal] / Role owner = [admin] / Role tenant = [*] / Role Id = [8280f869-29cd-4751-9303-fe0fd3e392d0]
# 6 / Role name = [Operator] / Role owner = [admin] / Role tenant = [*] / Role Id = [bed807c6-978c-4725-a5c0-8f6db22428d6]
# 7 / Role name = [Capacity Operator] / Role owner = [admin] / Role tenant = [*] / Role Id = [4cd2f14d-1575-4a31-9ea7-7f0c60247142]
```

getusergrouplist (TSPS / TrueSight)

Arguments

none

The getusergrouplist function shows all User Groups defined in the Presentation Server with the following information:

- The Group name

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getusergrouplist
```

Example Output:

```
# 1 / Group name = [Operators]
# 2 / Group name = [Central Monitoring Administrators]
# 3 / Group name = [API Group]
# 4 / Group name = [Viewers]
# 5 / Group name = [WS Full Access]
# 6 / Group name = [Capacity_Planning]
```

getdevicegrouplist (TSPS / TrueSight)

Arguments:

none

The getdevicegrouplist function shows all Device Groups defined in the Presentation Server with the following information:

- The Device Group name
- The Device Group Id
- The Component Type

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f getdevicegrouplist
```

Example Output:

```
# 1 / Device Group name = [ASSO_Cluster] / Device Group Id = [0,TS15] / Component Type = [bmcrealm]
# 2 / Device Group name = [BMC_TrueSight] / Device Group Id = [13,1] / Component Type = [BmcRealm]
...
Total # Device Groups = [15]
```

geteventgrouplist (TSPS / TrueSight)

Arguments:

none

The geteventgrouplist function shows all Event Groups defined in the Presentation Server with the following information:

- The Event Group name
- The Event Group Id

NOTE:

By default the `geteventgrouplist` command creates a structured view to reflect the Event Group structure. Use the `-tbl` and / or `-csv` option to get an unstructured output!

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f geteventgrouplist
```

Example Output:

```
Event Group Name = [Event Groups] / Event Group Id = [0]
--> Event Group Name = [AVM Events] / Event Group Id = [17]
--> Event Group Name = [All Active Events] / Event Group Id = [10001]
--> Event Group Name = [All Events] / Event Group Id = [10006]
--> --> Event Group Name = [TDS_Test_2] / Event Group Id = [3]
--> --> Event Group Name = [By Status] / Event Group Id = [10007]
--> Event Group Name = [BIT Test] / Event Group Id = [52]
--> Event Group Name = [By Status] / Event Group Id = [10012]
--> Event Group Name = [By User] / Event Group Id = [10009]
--> --> Event Group Name = [Unassigned] / Event Group Id = [10010]
--> Event Group Name = [Capacity Events] / Event Group Id = [10018]
--> Event Group Name = [Generated Incidents] / Event Group Id = [10024]
--> --> Event Group Name = [Event Incidents By Status] / Event Group Id = [10025]
--> --> Event Group Name = [Incident Errors] / Event Group Id = [10026]
--> Event Group Name = [IKEA Keynote Alerts] / Event Group Id = [1]
--> Event Group Name = [IT Data Analytics] / Event Group Id = [10020]
--> Event Group Name = [Information Events] / Event Group Id = [18]
--> Event Group Name = [Intelligent Events] / Event Group Id = [10003]
--> Event Group Name = [PATROL] / Event Group Id = [10015]
--> Event Group Name = [Related Events] / Event Group Id = [10014]
--> Event Group Name = [TDS] / Event Group Id = [11]
--> --> Event Group Name = [TDS_CUST1] / Event Group Id = [12]
--> --> --> Event Group Name = [CLOSE] / Event Group Id = [16]
--> Event Group Name = [Timos Dynamic Host Group] / Event Group Id = [6]
--> Event Group Name = [Truesight Infrastructure Health] / Event Group Id = [10021]
--> --> Event Group Name = [By Status] / Event Group Id = [10022]
--> Event Group Name = [ptf-sles-23] / Event Group Id = [9]

Total # Event Groups = [28]
```

deploypackage (TSPS / TrueSight)

Arguments:

- | | | |
|--------|----------|--|
| - pn | Required | Deployable Package name |
| - sid | Required | Server Id of the TSIM the Agent(s) are connected to |
| - aidl | Required | Comma separated list of Agent Ids that should be installed |
| - ima | Optional | Force install on systems with multiple Agents on |

The `deploypackage` function initiates the installation of the Deployable Package on the Agents referenced by the Agent Id and connected to the TSIM mentioned! To monitor the state of the deployment use the field "Deployment Status" of the `getagentlist` function!

Example Command:

```
trapic.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -f deploypackage
```

searchcis (TSPS / TrueSight)

Arguments:

- | | | |
|---------|----------|----------------------------------|
| - cif | Required | Filter to use when searching CIs |
| - cimax | Optional | Maximum number of results |

The searchcis function can be used to search for CIs and return a number of information about the resulting CIs like Name, Display name, McUdid, Server Id and Key.

Example Command:

```
tragic.exe -cs configs\TrueSight113.cfg -f searchcis -cif "ptf" -tbl
```

Example result:

```
# 1 / CI Name [ptf-sles-50] / CI Display Name = [ptf-sles-50 (BMC_ComputerSystem)] / CI McUdid = [19+ptf-sles-54_340] / CI Server Id = [19] / CI Key = [19+ptf-sles-54_340]
# 2 / CI Name [ptf-sles-51.truesight.tabuso.com] / CI Display Name = [ptf-sles-51.truesight.tabuso.com (BMC_ComputerSystem)] / CI McUdid = [19+ptf-sles-54_22] / CI Server Id = [19] / CI Key = [19+ptf-sles-54_22]
# 3 / CI Name [ptf-sles-52.truesight.tabuso.com] / CI Display Name = [ptf-sles-52.truesight.tabuso.com (BMC_ComputerSystem)] / CI McUdid = [19+ptf-sles-54_343] / CI Server Id = [19] / CI Key = [19+ptf-sles-54_343]
```

searchdevices (TSPS / TrueSight)

Arguments:

- | | | |
|---------|----------|--------------------------------------|
| - dif | Required | Filter to use when searching devices |
| - dimax | Optional | Maximum number of results |

The searchdevices function can be used to search for Devices and return a number of information about the resulting Devices like Name, Id, Number of Events and Agents, Disconnected Agents and Marked for delete.

Example Command:

```
tragic.pl -cs configs\TrueSight113.cfg -f searchdevices -dif "ptf-sles" -tbl -debug
```

Example result:

```
# 1 / Device ID = [250] / Device name = [ptf-sles-50] / Total Events = [3] / Total Agents = [1] / Disconnected Agents = [0] / Marked for delete? = [false]
# 2 / Device ID = [227] / Device name = [ptf-sles-51.truesight.tabuso.com] / Total Events = [13] / Total Agents = [1] / Disconnected Agents = [0] / Marked for delete? = [false]
# 3 / Device ID = [251] / Device name = [ptf-sles-52.truesight.tabuso.com] / Total Events = [2] / Total Agents = [2] / Disconnected Agents = [0] / Marked for delete? = [false]
```

Using TRAPIC II wrapper

Introduction

trapic wrappers are small scripts that utilize TRAPIC II to simplify a whole process or function. Most of them have the same connection arguments TRAPIC II itself has and can handle connection data files.

trgetdevicedata (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-dn [Device name]	The name of the device you want to get the data from
-mt [Monitor type name]	The name of the Monitor Type
-mi [Monitor instance name]	The name of the Monitor Instance
-at [The Attribute name]	The name of the Attribute
-time [Minutes]	Time frame from now into the past in minutes

trgetdevicedata gathers attribute raw data through the TSPS Rest-API from the TSIM database. It uses multiple trapic requests to translate the device name into the device id and the

Example Command:

```
trgetdevicedata.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -dn ptf-sles-10.truesight.tabuso.com -mt Swap -mi sda1 -at "Free Space" -time 10
```

Example Output:

```
# 1 / Data point date & time = [10.11.2018 16:28:08] / Data point timestamp (epoch) = [1541863688] / Data point value = [2.21]
# 2 / Data point date & time = [10.11.2018 16:29:08] / Data point timestamp (epoch) = [1541863748] / Data point value = [0.65]
# 3 / Data point date & time = [10.11.2018 16:30:09] / Data point timestamp (epoch) = [1541863809] / Data point value = [2.34]
# 4 / Data point date & time = [10.11.2018 16:31:09] / Data point timestamp (epoch) = [1541863869] / Data point value = [0.64]
# 5 / Data point date & time = [10.11.2018 16:32:09] / Data point timestamp (epoch) = [1541863929] / Data point value = [0.8]
# 6 / Data point date & time = [10.11.2018 16:33:09] / Data point timestamp (epoch) = [1541863989] / Data point value = [0.72]
# 7 / Data point date & time = [10.11.2018 16:34:09] / Data point timestamp (epoch) = [1541864049] / Data point value = [0.57]
# 8 / Data point date & time = [10.11.2018 16:35:09] / Data point timestamp (epoch) = [1541864109] / Data point value = [0.77]
# 9 / Data point date & time = [10.11.2018 16:36:09] / Data point timestamp (epoch) = [1541864169] / Data point value = [0.55]
# 10 / Data point date & time = [10.11.2018 16:37:09] / Data point timestamp (epoch) = [1541864229] / Data point value = [0.71]
```

trdevicecompare (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-dn1 [Device name]	The name of the device you want to get the data from
-dn2 [Monitor type name]	The name of the Monitor Type
-sca	Show common attributes

trdevicecompare gathers all monitor types, monitor instances and attributes from the two devices and checks if they are the same. Finally the command shows the attributes missing on device one and on device two. By adding the option -sca it also shows the attributes found on both devices.

Note:

Please note! The command is in beta state!

Example Command:

```
trdevicecompare.pl -u apiuser -p XYZXYZYZ -te BmcRealm -ts ptf-sles-9:8043 -dn1 ptf-sles-17.truesight.tabuso.com -dn2 ptf-sles-18.truesight.tabuso.com
(T)ruessight (R)est (API) (C)lient V2.2.00 (BETA) / Wrapper (Device compare)
(c) TAUBER Business Solutions UG
written by Timo Schmidt / http://www.tabuso.com/
Customer name: TAUBER and B / Max # session: none!
Expiration data 01.03.2019!
```

.....

Attributes on [ptf-sles-17.truesight.tabuso.com] but NOT on [ptf-sles-18.truesight.tabuso.com]:
Memory, Linux OS/Memory, Memory Used By User Processes and Kernel (Excludes Buffers-Cache)

Attributes on [ptf-sles-18.truesight.tabuso.com] but NOT on [ptf-sles-17.truesight.tabuso.com]:

traddservicetoprofile & trremoveservicefromprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presenation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-an [Authprofile name]	The name of the Authorization Profile you want to add the Service to.
-sn [Service name]	The name of the Service you want to add,

Traddservicetoprofile and trremoveservicefromprofile can be used to add and remove existing Services to and from an Authorization Profile to automate the process of maintaining Authorization Profiles.

Example Command:

```
traddservicetoprofile.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -an "TEST_PROFILE" -sn "Test Service 17"
```

traddashboardtoprofile & trremovedashboardfromprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-an [Authprofile name]	The name of the Authorization Profile you want to add the Service to.
-dashn [Dashboard name]	The name of the Shared Dashboard you want to add,

traddashboardtoprofile and trremovedashboardfromprofile can be used to add and remove existing Shared Dashboards to and from an Authorization Profile to automate the process of maintaining Authorization Profiles.

Example Command:

```
traddservicetoprofile.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -an "TEST_PROFILE" -dashn "Test Dashboard"
```

traddapplicationtoprofile & trremoveapplicationfromprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presenation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-an [Authprofile name]	The name of the Authorization Profile you want to add the Service to.
-appn [Application name]	The name of the Application you want to add,

traddapplicationtoprofile and trremoveapplicationfromprofile can be used to add and remove existing Applications to and from an Authorization Profile to automate the process of maintaining Authorization Profiles.

Example Command:

```
traddapplicationtoprofile.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -an "TEST_PROFILE" -appn "Test Application 17"
```

tradddevicetoprofile & trremovedevicefromprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presenation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-an [Authprofile name]	The name of the Authorization Profile you want to add the Service to.
-dn [Device name]	The name of the Device you want to add or remove to or from the Auth Profile.

tradddevicetoprofile and trremovedevicefromprofile can be used to add and remove existing Devices to and from an Authorization Profile to automate the process of maintaining Authorization Profiles.

Example Command:

```
tradddevicetoprofile.exe -u apiuser -p XYZXYZXYZ -te BmcRealm -ts ptf-sles-9:8043 -an "TEST_PROFILE" -dn "ptf-sles-11.truesight.tabuso.com"
```

tradddevicegrouptoprofile & trremovedevicegroupfromprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

The two commands add or remove an existing Device Group to or from an existing Authorization Profile!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-an [Authprofile name]	The name of the Authorization Profile you want to add the Service to.
-dgn [Device Group name]	The name of the Device Group you want to add or remove to and from the Auth Profile.

Example Command:

```
tradddevicegrouptoprofile.exe -u apiuser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -an "TEST PROFILE 2" -dgn "TEST GROUP"
```

```
trremovedevicegroupfromprofile.exe -u apiuser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -an "TEST PROFILE 2" -dgn "TEST GROUP"
```

traddeventgrouptoprofile & trremoveeventgroupfromprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

The two commands add or remove an existing Eventgroup to or from an existing Authorization Profile!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-an [Authprofile name]	The name of the Authorization Profile you want to add the Service to.
-egid [Eventgroup Id]	The Id of the Eventgroup you want to add or remove to or from the Auth Profile.

Example Command:

```
traddeventgrouptoprofile.exe -u apluser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -an "TEST PROFILE 2" -egid 10002
```

```
trremoveeventgroupfromprofile.exe -u apluser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -an "TEST PROFILE 2" -egid 10002
```

trcreateauthprofile & trdeleteauthprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-an [Authprofile name]	The name of the Authorization Profile you want to add the Service to.
-gn [Group name]	Comma separated list of Usergroups you want to add to the new Auth Profile
-rn [Role name]	Comma separated list of Roles you want to add to the new Auth Profile
-fac [Full Access List] (Optional)	List separated by pipe for areas you want to grant full access to, e.g. "SERVICE EVENTGROUP"
	MOPOLICY = Monitoring Policy Configuration
	SOLUTION = PATROL Solutions
	ACL = PATROL Agent ACLs
	DEVICE = Devices
	GROUP = Groups
	APPLICATION = Applications
	EVENTGROUP = Event Groups
	SERVICE = Services
	VIEW = Views
	MONITORGROUP = Monitor Groups
	COMPONENTFOLDER = Component Folders
	EVENTFOLDER = Event Folders

REPORT = Reports
SLO = SLOs
SHARED DASHBOARD = Shared Dashboards

`trcreateauthprofile` and `trdeleteauthprofile` can be used to add and remove Authorization Profiles to the Presentation Server.

Note:

- `trdeleteauthprofile` only needs the name of the Authorization Profile you want to remove

Example Command:

```
trcreateauthprofile.exe -u apiuser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -an "TEST PROFILE 2" -gn "Operators" -rn "Operator"
trdeleteauthprofile.exe -u apiuser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -an "TEST PROFILE 2"
```


tradddevicetoapplication & trremovedevicefromapplication (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presenation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-appn [Application name]	The name of the Application to add the Device to.
-dn [Device name]	The name of the Device you want to add to the Application.
-prop [ON OFF]	Switch propagation on or off for this Device.
-con [Node name]	Name of the Node to connect this Device to.

tradddevicetoapplication and trremovedevicefromapplication can be used to add and remove existing Devices to and from an Application to automate the process of maintaining Applications in TrueSight.

Example Command:

```
traddservicetoprofile.exe -u apiuser -p XYZXYZXYZ -te * -ts ptf-sles-9:8043 -appn "this-application" -dn "this-server.this-domain.com" -prop "ON"
```

trcopyauthprofile (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-ssys [Connection file]	Source TrueSight system connection file
-tsys [Connection file]	Target TrueSight system connection file
-san [Authprofile name]	Source Authprofile name
-tan [Authprofile name]	Name to copy profile to in target system
-f [validate copy]	Function
-debug	

trcopyauthprofile.exe can be used to copy Authprofile content within a TrueSight system or from one TrueSight system to the other. The tool can be run in two different modes using option “-f”. Using the validate function the tool just checks and displays if all basics (Role, Devices. etc.) exist in the target system.

If you use the function “copy” the tool will copy the following content of the Authorization Profile to the target system:

- Usergroups
- Roles
- Associated Objects
- Services
- Applications
- Groups
- Eventgroups
- Devices
- Shared Dashboards

Note: trcopyauthprofile.exe has to be used with Config files containing the connection data! Using the username / password etc. arguments is not supported!

Example Command:

```
trcopyauthprofile.pl -ssys configs\TrueSight113.cfg -tsys configs\TrueSight113.cfg -san "NEW_TESTPROFILE" -tan "NEW_TESTPROFILE2" -f validate -debug
```

trdeletemanageddevice (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-n [Managed Device]	Name of the Managed Device to delete
-po [Port]	Port of the Managed Device to delete
-setIs [IS FQDN]:[IS ServerPort]	
(Optional)	Integration Service to delete the Agent from. This reduces the REST-API traffic because the tool does not have to retrieve all PATROL Agents!

trdeletemanageddevice can be used to remove a disconnected Managed Devices (=PATROL Agent) from the TrueSight system.

Note:

- Make sure the Managed Device is disconnected

Example Command:

```
trdeletemanageddevice.exe -u apluser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -n testserver.tabuso.com -p 3181
```

trdeploypackage (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-anl [Agent list]	Comma separated list of hostnames, FQDNs or IP-Addresses of PATROL Agents to deploy to.
-anlf [Agent list file]	File with comma separated list of Agents.
-dpm [Deployable Package]	Name of the Deployable Package

trdeploypackage can be used to initiate installations of Deployable Packages on one or more PATROL Agents.

Example Command:

```
trdeploypackage.exe -u apiuser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -anl "patrolagentserver" -dpm "WindowsX64_Package"
```

trcreateapplication & trdeleteapplication (TSPS / TrueSight)

IMPORTANT: Please make sure that you test the tool against a representing test or development system before running the tool in your production environment!

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
	Note: The uber tenant is
	BmcRealm <= 10.7.xx
	* > 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-appn [Application name]	Name of the Application to create
-desc [Description]	Description of the Application
-syn [ON OFF]	Synthetic on or off
-ici [CI Name]	Name of the CI to impact

trcreateapplication.exe creates the Application with the options provided in TSPS and – if available – AVM Portal. trdeleteapplication.exe will delete the Application provided by the name.

Example Command:

```
trcreateapplication.exe -cs configs\TrueSight113.cfg -appn "ToolCreated" -desc "TEST TEST TES" -syn OFF -imp MEDIUM
```

trrestartagent (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-n [Agent name]	Name of the Agent to get Agent Setup from
-po [Port]	Port of the Agent
-nowarn	Suppress Monitoring Studio warning

IMPORTANT:

The Monitoring Studio TSPS Component from Sentry Software must be installed in the TrueSight Presentation Server!

trrestartagent.exe restarts the PATROL Agent given with -n and -po.

Example Command:

```
trrestartagent.exe -cs configs\TrueSight113.cfg -n "ptf-sles-50" -po 3181
```


trgetagentsetup (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-n [Agent name]	Name of the Agent to get Agent Setup from
-po [Port]	Port of the Agent
-def	Get with Defaults
-nowarn	Suppress Monitoring Studio warning

IMPORTANT:

The Monitoring Studio TSPS Component from Sentry Software must be installed in the TrueSight Presentation Server!

trgetagentsetup.exe retrieves the PATROL Agent Setup from the Agent provided.

Example Command:

```
trgetagentsetup.exe -cs configs\TrueSight113.cfg -n "ptf-sles-50" -po 3181 -def
```


trsetagentsetup (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-n [Agent name]	Name of the Agent to get Agent Setup from
-po [Port]	Port of the Agent
-vp	Path to the AgentSetup folder e.g. "/TEST/TEST"
-vs	Variables in JSON format Variable / Value pairs e.g. {"testVariable\":"testValue\"}"
-nowarn	Suppress Monitoring Studio warning

IMPORTANT:

The Monitoring Studio TSPS Component from Sentry Software must be installed in the TrueSight Presentation Server!

trsetagentsetup.exe sets Variables and values in the PATROL Agent Setup of the Agent provided.

Example Command:

```
trsetagentsetup.exe -cs configs\TrueSight113.cfg -n "ptf-sles-50" -po 3181 -vp "/TEST/TEST/" -vs '{"testVariable\":"testValue\"}'
```

trdeleteagentsetup (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-n [Agent name]	Name of the Agent to get Agent Setup from
-po [Port]	Port of the Agent
-vp	Path to the AgentSetup Variable e.g. "/TEST/TEST/testVariable"
-nowarn	Suppress Monitoring Studio warning

IMPORTANT:

The Monitoring Studio TSPS Component from Sentry Software must be installed in the TrueSight Presentation Server!

trdeleteagentsetup.exe deletes Variables and values in the PATROL Agent Setup of the Agent provided.

Example Command:

```
trdeleteagentsetup.exe -cs configs\TrueSight113.cfg -n "ptf-sles-50" -po 3181 -vp "/TEST/TEST/testVariable"
```

trgetlicensedata (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-tsim [TSIM FQDN ALL]	Name of the TSIM Server to get the license data for or "ALL" for all Servers

trgetlicensedata.exe retrieves the license data for the TSIM Server or when "ALL" is provided for the whole TrueSight System.

Example Command:

```
trgetlicensedata.exe -cs configs\TrueSight113.cfg -tsim ALL
```

trexportpolicy (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-pn [Policyname]	Name of the Policy to export
-npn [New Policyname]	JSON File will be changed so if you import the file as a new Policy it will have the new name
-fn [Filename]	Name of the file the Policy content in JSON format should be written to

trexportpolicy.exe retrieves the Policy and writes the content to the file provided in JSON format. Optionally the name of the Policy in the file can be changed so when reimporting the Policy will have the new name.

Example Command:

```
trexportpolicy.exe -cs configs\TrueSight113.cfg -pn "LX_LINUX_KM" -npn "NEW_LINUX_KM" -fn NEW_LINUX_KM.policy
```

trimportpolicy (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-fn [Filename]	Filename of the Policy content in JSON format that should be imported.
-or	Override Policy if it already exists

trimportpolicy.exe imports the Policy provided in JSON Format. Optionally an existing Policy can be overwritten (= delete and create new).

Example Command:

```
trimportpolicy.exe -cs configs\TrueSight113.cfg -fn NEW_LINUX_KM.policy
```

trupdatepolicy (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-fn [Filename]	Filename of the Policy content in JSON format that should be imported.

trupdatepolicy.exe updates the Policy provided in JSON Format.

Example Command:

```
trupdatepolicy.exe -cs configs\TrueSight113.cfg -fn NEW_LINUX_KM.policy
```

trmodifypolicy (TSPS / TrueSight)

Arguments:

- u [Username] The name of the user to use for the authentication
- p [Password] The password of the user
- te [Tenant] The tenant you want to authenticate at
- Note: The uber tenant is
- BmcRealm <= 10.7.xx
- * > 10.7.xx
- ts [Server]:[Port] The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
- ss [Filename] Store connection data in file
- cs [Filename] Connect to system stored in file
- pn [Policyname] Name of the Policy to modify.
- act [enable||disable||share] Action to execute on the Policy provided

trmodifypolicy.exe enables, disables or shares the Policy provided by name.

Example Command:

```
trmodifypolicy.exe -cs configs\TrueSight113.cfg -pn "NEW_LINUX_KM" -act "enable"
```

trdeletepolicy (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at

Note: The uber tenant is

BmcRealm <= 10.7.xx

* > 10.7.xx

-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
---------------------	---

-ss [Filename] Store connection data in file

-cs [Filename] Connect to system stored in file

-pn [Policyname] Name of the Policy to modify.

trdeletepolicy.exe deletes the Policy provided by name.

Example Command:

```
trdeletepolicy.exe -cs configs\TrueSight113.cfg -pn "NEW_LINUX_KM"
```


traudit (TSPS / TrueSight)

Arguments:

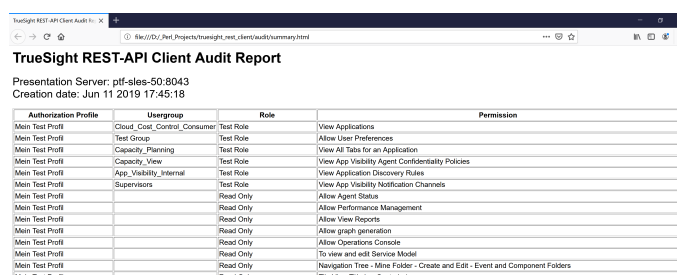
- u [Username] The name of the user to use for the authentication
- p [Password] The password of the user
- te [Tenant] The tenant you want to authenticate at
- Note: The uber tenant is
 BmcRealm <= 10.7.xx
 * > 10.7.xx
- ts [Server]:[Port] The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
- ss [Filename] Store connection data in file
- cs [Filename] Connect to system stored in file
- out [Path] Path where the report file should be created
- html Create the report in HTML format
- batch No request for confirmation before executing

traudit can be used to create an audit report called “summary.csv” or “summary.html” containing all Authorization Profiles, Usergroups, Roles and Permissions in one single report file. The file is generated with header in comma separated format by default. Using the “-html” option it generates the report in HTML format.

Using the “-batch” option the tool does not ask for a confirmation before generating the report!

Example Command:

```
traudit.exe -u apiuser -p XYZXYZ -te * -ts ptf-sles-50.truesight.tabuso.com:8043 -out "audit" -html -batch
```



Authorization Profile	Usergroup	Role	Permission
Main Test Profil	Cloud_Cost_Control_Consumer	Test Role	View Applications
Main Test Profil	Test Group	Test Role	Allow User Preferences
Main Test Profil	Capacity_Planning	Test Role	View All Tabs for an Application
Main Test Profil	Capacity_View	Test Role	View App Visibility Agent Confidentiality Policies
Main Test Profil	App_Visibility_Internal	Test Role	View Application Discovery Rules
Main Test Profil	Supervisors	Test Role	View App Visibility Notification Channels
Main Test Profil		Read Only	Allow Agent Status
Main Test Profil		Read Only	Allow Performance Management
Main Test Profil		Read Only	Allow View Reports
Main Test Profil		Read Only	Allow graph generation
Main Test Profil		Read Only	Allow Operations Console
Main Test Profil		Read Only	To view and edit Service Model
Main Test Profil		Read Only	Navigation Tree - Mine Folder - Create and Edit - Event and Component Folders
Main Test Profil		Read Only	Title View Title bar Controls Area

Pic. 1: HTML Audit report summary.html

trexportdashboards (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-dnl [Dashboardnamelist]	List of Dashboards or "ALL" for all Dashboards to export
-tdn [Target Directory name]	Complete path to the directory where the Dashboard JSON files should be stored
-ffn	Fix File names / Fixes the most common problems of characters allowed in Dashboard names but not in Filenames
-iafn	ID as Filename / Stored the Dashboard in a file named with the ID of the Dashboard.

trexportdashboards.exe exports all Dashboards the connecting User can see into single files in the directory give. Each Dashboard is stored as JSON.

Example Command:

```
trexportdashboards.exe -cs configs\TrueSight11304.cfg -dnl "Timos Demo Dashboard" -tdn "c:\temp\_export"
trexportdashboards.exe -cs configs\TrueSight11304.cfg -dnl "ALL" -tdn "c:\temp\_export" -ffn
```

trimportdashboards.exe (TSPS / TrueSight)

Arguments:

-u [Username]	The name of the user to use for the authentication
-p [Password]	The password of the user
-te [Tenant]	The tenant you want to authenticate at
Note: The uber tenant is	
BmcRealm	<= 10.7.xx
*	> 10.7.xx
-ts [Server]:[Port]	The Rest-API of the TrueSight Presentation Server in [Hostname]:[Port] format
-ss [Filename]	Store connection data in file
-cs [Filename]	Connect to system stored in file
-fni [Filename list]	List of exported Dashboard JSON Files that should be imported or "ALL" to import all Files in the Directory one after the other.
-tdn [Target Directory name]	Full Path to the directory where the Dashboard JSON files are stored
-fovr	Force Override / Forces to recreate and override the Dashboard also if the internal Dashboard ID has been changed after the export

trimportpolicy.exe imports Dashboards from an exported Dashboard JSON file. The Dashboard will be created if it does not exist.

IMPORTANT: The Dashboard will be owned by the User used to connect to the TSPS after the import!

Example Command:

```
trimportdashboards.exe -cs configs\TrueSight11304.cfg -fni "Timos Demo Dashboard.json" -tdn "c:\temp\_export"
trimportdashboards.exe -cs configs\TrueSight11304.cfg -fni "ALL" -tdn "c:\temp\_export" -fovr
```

Trouble shooting

If a command creates no or wired output

1.) Add the the “-debug” option

Check if the connection can be established or not or search the debug for error messages!

2.) Add the “-kof” option

This will keep the output of the REST-API request in the file trapic.exe.out. Check the file for error messages!

3.) Contact support@tabuso.com

To trouble shoot problems please add the “-debug” option and if you have a support contract please send the following information to support@tabuso.com:

- Problemdescription
- TrueSight Presentation Server and / or RSSO version
- Debugoutput from trapic.exe or trgetperfddata.exe

Known problems

No problems known yet!

ID	Status	Description
0000001	Open	Double index # in getmontypes
0000002	Open	trdevicecompare.exe shows monitor instances containing “()” as missing on both devices!