

**TeamCall Telephony Access Platform
TeamCall CSTA Server**

**Reference Guide for
Developers and Administrators**

TeamCall is a registered trademark of ilink Kommunikationssysteme GmbH (ilink).

All other trademarks, service marks, and trade names mentioned in this publication are owned by their respective owners.

While the information in this document is believed to be accurate, ilink makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

ilink shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance, or use of this document.

COPYRIGHT NOTICE

No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of ilink. No third-party intellectual property right liability is assumed with respect to the use of the information contained herein. ilink assumes no responsibility for errors or omissions contained in this document. This document and the features described herein are subject to change without notice.

Copyright © 2000-2023 by ilink Kommunikationssysteme GmbH. All rights reserved.

This manual describes TeamCall Server version 5.1.
May 2023.

ilink Kommunikationssysteme GmbH
Kurfürstendamm 67
10707 Berlin, Germany
www.ilink.de

Contents

Introduction	8
TeamCall Server	8
Document conventions.....	9
How this book is organized.....	9
Configuration.....	10
Main configuration file entries.....	10
callLogEnabled	10
cstaLinkAddress	11
cstaLinkPort	11
cstaLogEnabled	11
debugLevel	11
deviceMonitorConfFile	12
implicitAssociation	12
interfaceLogEnabled	12
license	12
logDir	13
logFileMaxBackups.....	13
logFileMaxSize.....	13
loginPort	13
maxPendingRequests	14
PBXPassword	15
pbxType	15
PBXUsername	15
prefixPlan.....	15
requestTimeout	16
useLocalTimeZoneTimeStamps.....	16
useNumericTimestamps	16
Example main configuration file.....	17
Device configuration file	17
Important notice	18
Starting and stopping TeamCall Server	19
Starting TeamCall Server.....	19
Possible startup scenarios	19
Example session: Starting TeamCall Server.....	21
Stopping TeamCall Server.....	22

Administering TeamCall Server using a SuperVisor session23

Viewing the current configuration	24
Activating the changes of a configuration file	24
Viewing license information.....	25
Viewing supported requests.....	25
Logging	26
The log files	26
Temporary changing path for log files	27
Enable/disable logging	27
Debugging modes.....	27
Monitoring Devices	28
Adding/removing new devices to be monitored "stickily"	28
More about device monitoring	28
Ending the SuperVisor session.....	30
Error messages.....	30
FILEERROR.....	30
ILINK-INTERNAL DEMOVERSION	30
INTERNALERROR	30
INVALCMD	30
INVALNUMPARAM	30
INVALPARAM	31
LICENSEERROR.....	31
LINKOUTOFSERVICE	31
NOMULTIPLEINSTANCE	31
NOTREGISTERED.....	31
PENDINGREQUEST	32
SUCCESS	32
UNAVAILBLEREQUEST	32

APIs 33

CSTA	33
JTAPI.....	33
STLI	33
TAPI.....	34

Log files 35

Activate logging.....	35
Sys.log	36
Status	36
Warnings (WRNXXXX)	37
WRN0003	37
WRN0004	37
WRN0005	37
WRN0006	37
WRN0007	37
WRN0009	37
WRN0010	38
WRN0011	38

WRN0012	38
WRN0013	38
Errors (ERRXXXX).....	38
ERR0001	38
ERR0002	39
ERR0003	39
ERR0004	39
ERR0005	39
ERR0006	39
ERR0009	39
ERR0010	40
ERR0011	40
ERR0012	40
ERR0013	40
ERR0014	40
ERR0015	41
ERR0016	41
ERR0017	41
ERR0018	41
ERR0019	41
ERR0020	41
ERR0026	42
ERR0030	42
ERR0031	42

Troubleshooting..... 43

Issues	43
After starting TeamCall Server, no connection to the phone system is established.....	43
Local devices cause an error, when accessed with STLI requests.....	44

1

Introduction

Welcome to ilink TeamCall Server.

TeamCall is the open Computer Telephony Integration (CTI) environment from ilink. Because of its open architecture, other developers can deploy TeamCall to use professional telephony features in their applications. The middleware layer of ilink TeamCall Server enables a simple and platform independent communication between CSTA interfaces of phone systems from vendors like Aastra, Alcatel, Avaya, Cisco, DeTeWe, Deutsche Telekom, Ericsson, Innovaphone, Mitel, Nortel, Panasonic, ShoreTel, Siemens, Tenovis, and Unify.

The TeamCall Server Development Kit supplies developers with all necessary information and tools to enable their applications to communicate with TeamCall Server.

TeamCall Server

TeamCall Server is the intelligent and complete solution for controlling a growing range of phone systems. It perfectly connects any CTI application to a phone system by translating telephony commands into the language of phone systems: CSTA (Computer Supported Telecommunication Applications).

CSTA is an industry standard, described by ECMA, an international Europe-based industry association founded in 1961 and dedicated to the standardization of information and communication systems. For more information visit their web site at www.ecma.ch. For information on TAPI, please refer to www.microsoft.com.

For CTI application development, TeamCall Server provides a powerful set of APIs, which allow you to build your own new CTI applications, increase features of existing ones, or test new concepts in a rapid prototyping process: STLI (Simple Telephony Interface), TAPI, JTAPI, and CSTA.

The connection between TeamCall Server and a phone system will be established using a network socket connection.

This allows just-in-time development of CTI applications as well as installation and modification of CTI installations like call centers.

Document conventions

This manual uses the following typographical conventions:

The `monospace font` in body text represents portions of code and names of items, commands, requests and keywords.

Whole paragraphs in `monospace font` represent samples of code. In examples, user input is formatted in **bold type**, on-screen output in `standard fontstyle`.

Variables are placed in `<>`.

Optional parameters are placed in `[]`.

Mutually exclusive parameter options are separated by `|`.

How this book is organized

- Chapter 1: Introduction
This chapter provides a general overview of TeamCall Server.
- Chapter 2: Configuration
A description of TeamCall Server's configuration files.
- Chapter 3: Starting and Stopping TeamCall
Describes how to start and stop TeamCall Server.
- Chapter 4: SuperVisor Interface
Describes how to make changes to the setup while TeamCall Server is running, modification of logging behavior and debugging mode.
- Chapter 5: APIs
Describes the APIs offered by TeamCall.
- Chapter 6: Log files
A list of logging files that are created when TeamCall Server is running.
- Chapter 7: Troubleshooting
Describes how to solve certain problems when running TeamCall Server.

2

Configuration

TeamCall Server uses two different configuration files:

`Default.conf` – The main configuration file. Every TeamCall Server software package contains this configuration file. It includes basic settings for the configuration of TeamCall Server. This file should be used as a source for generating your own new configuration files.

`DeviceLicense.conf` – Allows to limit the devices that may be monitored.

`DeviceMonitor.conf` – Device configuration file for permanently monitored devices. A device can be monitored from every startup until shut down of TeamCall Server by adding it to `DeviceMonitor.conf`. The path to this configuration file is set within the main configuration file using the key `deviceMonitorConfFile`.

`DialPlan.conf` – Optional rewriting of signaled phone numbers.

Main configuration file entries

This section describes all entries of the configuration file (`Default.conf`) sorted in alphabetical order.

callLogEnabled

Key	<code>callLogEnabled <0/1></code>
Example	<code>callLogEnabled 1</code>
Description	Specifies whether calls shall be logged or not. 0 turns logging off, 1 turns logging on. Default value is 0.

cstaLinkAddress

- Key** `cstaLinkAddress <TCP/IP address>`
- Example** `cstaLinkAddress 192.168.1.1`
- Description** Specifies the TCP/IP address of the phone system. If the `cstaLinkAddress` is not specified or the key does not exist, the default value will be used. The default value is 127.0.0.1 (localhost).
- If the `cstaLinkAddress` has not been specified in the main configuration file, it may be specified by starting TeamCall Server using the `-a` option. For further details on TeamCall Server command line parameters see *Starting and stopping TeamCall Server* on page 19.

cstaLinkPort

- Key** `cstaLinkPort <TCP/IP port number>`
- Example** `cstaLinkPort 2555`
- Description** Specifies the port number for TCP/IP socket connection. If the port number is not specified or the key does not exist, the default value will be used. The default value is 2555.

cstaLogEnabled

- Key** `cstaLogEnabled <0/1>`
- Example** `cstaLogEnabled 1`
- Description** Specifies whether plain CSTA messages shall be logged or not. 0 turns logging off, 1 turns logging on. Default value is 0.

debugLevel

- Key** `debugLevel <0-9>`
- Example** `debugLevel 0`
- Description** Specifies the debug level at which TeamCall Server will be started. The default value for running TeamCall Server is 0. For different levels of debugging, the `debugLevel` may be set to values between 0 and 9. If `debugLevel` has not been specified or the key does not exist, the default value will be used.
- Each debug level includes all other debug levels with a lower value.
- Possible values for the `debugLevel` are:
- 0: enable logging for performance analysis
 - 1: allow logging of CSTA calls
 - 2: allow logging of devices (adding and removing)
 - 5: ASN1 message decoding and logging (CSTAxxxAdaptor)
 - 6: system status messages, e.g., `link down` (CSTAxxxAdaptor)
 - 7: messages between model and adaptor (CSTAController)
 - 8: protocol trace between adaptor and PBX (CSTAxxxAdaptor)
 - 9: messages between adaptor and PBX (CSTAxxxAdaptor)

deviceMonitorConfFile

Key	deviceMonitorConfFile <absolutePath>
Example	deviceMonitorConfFile /opt/teamcall/DeviceMonitor.conf
Description	<p>Specifies the name and path of the device configuration file, which specifies the devices to be monitored. An absolute path should be specified. The default value for this key is</p> <pre><installationPath>/DeviceMonitor.conf</pre> <p>There are three ways to monitor a device at the phone system:</p> <p>A device can be monitored from every startup of TeamCall Server until shut down by adding it to the device configuration file. It also can be added in a SuperVisor session for the lifetime of the currently running TeamCall Server or it can be added temporarily by calling the appropriate requests in an STLI session. For further details, see <i>Device configuration file</i> on page 17.</p>

implicitAssociation

Key	implicitAssociation <0/1>
Example	implicitAssociation 1
Description	<p>Allows to specify if a phone system connection uses ACSE for connection establishment, or not. The second case is called <i>implicit association</i> and is selected when this parameter is set to the value 1.</p> <p>The default value, if not configured, is 0 (= no implicit association).</p> <p>This parameter is only used for pbxTypes <code>generic</code>, <code>hipath4000</code>, and <code>panasonic</code>.</p> <p>For example, a connection to Mitel/Aastra 400 phone systems using the pbxType <code>generic</code> requires to set <code>implicitAssociation</code> to 1.</p>

interfaceLogEnabled

Key	interfaceLogEnabled <0/1>
Example	interfaceLogEnabled 1
Description	<p>Specifies whether the network communication on CSTAServer's interfaces shall be logged separately or not. 0 turns logging off, 1 turns logging on. Default value is 0.</p>

license

Key	license <licenseKey>
Example	<pre>license 3ac6d7af928aeda3d7c40fff4e1d12a42f094de7a61d72 07e0d7bf4008fabae6bb17c3975b99abc1137d5cf3a253e141c219185307dc4c2e2 c50e0f2111ca419c806ad72b014c0f93960609f9ea26164cb4c51bb9978e8786235 2c1580a2419032738cc8frz5394b49</pre>
Description	<p>Specifies the license key for TeamCall Server. There is no default value. If no license key is specified, TeamCall Server will run in demo mode.</p>

logDir

Key logDir <directory>

Example logDir /tmp/logs

Description Directory where to log to, prepended to logfiles. If logDir has not been specified, the logfiles will be created within the default path. The default path for logging is the installation path.

If the path specified in logDir does not exist or is not writeable, or if the key logDir does not exist in the main configuration file, TeamCall Server log files will be created in the installation path of TeamCall Server.

logFileMaxBackups

Key logFileMaxBackups <0-n>

Example logFileMaxBackups 1

Description Number of backup files per log file. Default is 1 (one log file and one backup file). If the value of logFileMaxBackups is 0, no backups will be made. After reaching logFileMaxSize, the current log file will be deleted, newly created, then logging starts again.

logFileMaxSize

Key logFileMaxSize <Kbytes/-1>

Example logFileMaxSize 300

Description Maximum size of a log file in Kbytes. The default value is 100 Kbytes.

If $0 \leq \text{logFileMaxSize} \leq 100$ Kbytes, the value will be set to 100 Kbytes.

If the value of logFileMaxSize is -1, there will be no maximum limit for the size of the log file. It is up to the administrator to control the size of the log file.

If the value of logFileMaxSize is -1 no backup file will be created.

loginPort

Key loginPort <TCP/IP port number>

Example telnet CTIHost 26535
Trying x.x.x.x...
Connected to CTIHost.
Escape character is '^]'.
STLI
error_ind SUCCESS STLI

Description To start a certain session, a loginPort has to be used. After establishing a login session, the user decides what session type to start. When a session has been started, it is kept up during the connection. To start a different session, the login session has to be initiated again.

For example, after starting a login session via telnet, one of these three sessions can be started by issuing the appropriate command:

STLI – starts a STLI session, see *STLI Reference Guide*.

SuperVisor – starts a SuperVisor session, see chapter 4.

NetTSPI – starts a NetTSPI/TAPI session (not further documented).

A correct input results in the following response:

```
error_ind SUCCESS <Session>
```

The user must wait for a positive response before new commands can be issued. The login session can be closed with the command **BYE**. The default value for `loginPort` is 26535.

maxPendingRequests

Key `maxPendingRequests <seconds>`

Example `maxPendingRequests 27`

Description Key to determine the length of the list of pending requests. Default is 20; minimum is 20, maximum is 1000.

Every request which is sent from TeamCall Server (via the session) to the phone system is usually followed with a response (indication) from the phone system.

Example in STLI:

```
MakeCall 100 101
error_ind SUCCESS MakeCall
```

There is always the possibility that the phone system can no longer handle the number of requests and

- a response is sent very late or
- no response is sent at all.

To avoid this, additional requests from the session are put on a waiting list to take some pressure off the phone system.

All requests from the sessions are received by a central object (CSTAController) of TeamCall Server and then passed on to the phone system. Every request is put on a list of pending requests. After a response was received (`error_ind SUCCESS`, `error_ind UniversalFailure`, etc.) the request is taken off the list. If the list of pending requests has reached a certain size which is set by the key `maxPendingRequests`, additional requests are temporarily stored in another list, which is called the list of queued requests. After a request got a response and was removed from the list of pending requests, the next request from the list of queued requests is passed on.

The length of the list of queued requests is not limited. The length of the list of pending requests is set with this key. See also key `requestTimeout`.

Note: The value for `maxPendingRequests` should not be set too high. Tests with Siemens Hicom150 and Alcatel 4400 resulted in requests not being answered or rejected after a certain number of requests.

PBXPassword

- Key** PBXPassword <password of phone system user>
- Example** PBXPassword 895pQ2111
- Description** Specifies the password of a phone system user for cases where the phone system requires authentication on its CTI port. See also PBXUsername.

pbxType

- Key** pbxType <phone system type>
- Example** pbxType cisco
- Description** Specifies the type of the phone system that the server shall connect to. If this is not configured, the phone system type needs to be specified via the command line parameter `-pbxType` when `CSTAServer` is started.

Note: List possible values for `-pbxType` by calling `CSTAServer -l` (lower case L)

Note: See configuration parameter `implicitAssociation` when using the phone system type `generic` for Mitel/Aastra 400 phone systems.

PBXUsername

- Key** PBXUsername <username of phone system user>
- Example** PBXUsername ctiuser
- Description** Specifies the username of a phone system user for cases where the phone system requires authentication on its CTI port. See also PBXPassword.

prefixPlan

- Key** prefixPlan "<pattern>/<replacement>?[:<pattern>/<replacement>]*"
- Example** prefixPlan "+90/90:+9/9:+/90"
- Description** Specifies replacement rules for leading strings of dialing numbers in call events. A rule consists of two strings, separated by a slash character: the string to be replaced and the replacing string. Rules are separated by colons. This defines how the phone system reports all numbers of calling and called parties.

requestTimeout

Key requestTimeout <seconds>

Example requestTimeout 30

Description Key to set the maximum time in seconds which a request stays on the pending or queued list. The default value is 30, the minimum value is 20 and the maximum is -1 (unlimited).

To avoid having requests on the pending or queued list forever if the phone system is overloaded, there is a time limit for requests on both lists which is set by the key `requestTimeout`. Both lists are constantly monitored and if a request has reached the time set with `requestTimeout`, it is removed. In this case, the following error message is sent to the session:

```
error_ind REQUESTTIMEOUT <Request>
```

See also key `maxPendingRequests`.

useLocalTimeZoneTimeStamps

Key useLocalTimeZoneTimeStamps <0/1>

Example useLocalTimeZoneTimeStamps 0

Description Specifies the time zone used in log file timestamps. If set to 1, timestamps use the local time zone. If set to 0, timestamps use UTC.

The default value is 0 (use UTC).

Note: This parameter is not used if `useNumericTimestamps` is set to 1 (use numeric format).

useNumericTimestamps

Key useNumericTimestamps <0/1>

Example useNumericTimestamps 0

Description Specifies the format of log file timestamps. If set to 1, the time stamps are printed in a numeric format as seconds since Jan 1, 2001 00:00:00 UTC. E.g., 698167891.626737

If set to 0, time stamps are printed in text format. E.g., 2023-02-15 11:45:45.569169

The default value is 0 (use text format).

Example main configuration file

```
##### TCP/IP Addresses and Ports
# TCP address of PBX
cstaLinkAddress 10.1.1.1
# Port of PBX
cstaLinkPort    2555
# TCP Port for login
loginPort      19000
##### Logging
logDir /tmp/debug
#directory where to log to, prepended to logfiles
callLogEnabled 1
#0=false, 1=true ; logs to callLogFile
cstaLogEnabled 1
#0=false, 1=true ; logs to cstaLogFile.
debugLevel    9
logFileMaxBackups 3
logFileMaxSize 250
##### Initial Monitoring
deviceMonitorConfFile /opt/pbx//DeviceMonitorDebug.conf #should be
absolute path
##### Replacement Rules for replacing leading strings in Dialing
Numbers
prefixPlan "+00/00:+0/0:+/00"
```

Device configuration file

When starting TeamCall Server, the device configuration file containing device numbers will be read (`DeviceMonitor.conf`). These devices are monitored from startup of TeamCall Server. No further monitor request is necessary.

Note: The initially monitored devices may be removed at runtime via a SuperVisor session using the request:
`RemoveDeviceMonitor <device>`

This request does not modify the device configuration file. For further details on `RemoveDeviceMonitor` see *Monitoring Devices* on page 28.

Every device connected to a phone system has a unique internal number. The device configuration file must contain an entry for each device, a monitoring point shall be set for. The following restrictions apply to this file:

- one device number per line
- no blank lines in between
- a new line after the last entry
- no remarks

Example

```
111
185
123
168
```

The device entries can be placed in any order.

Refer to *More about device monitoring* on page 28 for details.

The name of this file (default: `DeviceMonitor.conf`) must be specified using its absolute path as a value of the key `DeviceMonitorConfFile` in the main configuration file.

Every valid device listed in the device configuration file will be monitored from startup until shut down of TeamCall Server. Further devices may be monitored by using the appropriate requests within different sessions. See *Monitoring Devices* on page 28.

These devices will only be monitored as long as at least one session is running.

Important notice

Please note that it may take a while before a monitoring point has been added to a device. For every entry in the device configuration file a request is sent to the phone system. The response time until these requests are acknowledged depends on many circumstances:

- network load
- load of the phone system
- load of the TeamCall Server computer

Due to too much traffic, it may be possible that the responses of the phone system will be delayed or not sent at all. This could cause synchronization problems.

As long as a monitoring point is not active, no event reports for this device can be received.

Additionally, no sessions will be able to add a monitoring point to this device (an attempt gets the response `error_ind PENDINGREQUEST`).

The progress of adding monitoring points can be supervised from the SuperVisor session with the request `ShowDeviceMonitors`, see section *More about device monitoring* on page 28.

For adding a greater number of monitoring points (more than 40), we recommend using the request `AddDeviceMonitor` from within the SuperVisor session. After issuing each `AddDeviceMonitor` request you should wait to get the response of the phone system. This way you make sure that monitoring points will be added correctly in the right order. Also, you'll receive feedback from the phone system if it gets too busy. In such a case any further command will result in a negative response.

3

Starting and stopping TeamCall Server

In the following section you will find the command line options and default settings for starting TeamCall Server.

The default settings allow to run TeamCall Server without modifying the shipped TeamCall Server software package.

Starting TeamCall Server

Syntax for all operating systems:

```
CSTAServer [-a <TCP/IP address>] [-c <configFile>] [-pbxType <pbxType>]
```

Additional options for Microsoft Windows:

```
[ -i | -u | -s ]
```

Additional option for Linux systems:

```
[-d]
```

Possible startup scenarios

Syntax CSTAServer [-pbxType <pbxType>]

Description Starting TeamCall Server with default settings. In this case, TeamCall Server looks for the main configuration file `Default.conf` in its installation path (i.e., where the TeamCall Server binary is installed).

Use the `-pbxType` switch to specify the type of phone system you are using if this is not configured via the configuration parameter `implicitAssociation` in `Default.conf`.

Note: List possible values for `-pbxType` by calling `CSTAServer -l` (lower case L)
List information about available command line parameters by calling `CSTAServer -h`

If the main configuration file was not found, TeamCall Server terminates with an error message.

If the main configuration file was found, TeamCall Server starts up with the settings loaded from this file.

Note: The default main configuration file shipped with the TeamCall Serversoftware package has no entries set for the TCP/IP address of the phone system and the device configuration file (in fact these entries are inserted as remarks). To run TeamCall Server without making modifications, it must be started with the command line option `-a` (see next section).

The device configuration file contains entries for all devices which are to be permanently monitored while TeamCall Server is running.

If the device configuration file does not exist (or is empty), no permanent monitoring points for monitoring devices will be set. In this case, monitoring points can be added as follows:

- in a "sticky" way, using the `AddDeviceMonitor` request or
- by sending a `MonitorStart` request from an STLI session (see *MonitorStart* in the STLI Reference Guide).

If `AddDeviceMonitor` has been used, the specified device will be monitored until TeamCall Server is shut down or until the monitoring will be stopped using the `RemoveDeviceMonitor` request. In the other case, state changes of the specified device will be traced as long as this device is monitored in at least one STLI session.

If the main configuration file exists, but does not have entries for `cstaLinkPort` or `loginPort`, the following internal default values will be used:

```
cstaLinkPort      2555
loginPort         26535
```

Syntax `CSTAServer -a <cstaLink TCP/IP address> [-pbxType <pbxType>]`

Description TeamCall Server receives the TCP/IP address of the phone system from the command line. If `<cstaLink TCP/IP Address>` is invalid, TeamCall Server will terminate with an error message. All other values are loaded from the main configuration file. If it does not exist, TeamCall Server will start using the default values.

```
cstaLinkPort 2555
loginPort 26535
```

Syntax `CSTAServer -c <configFile> [-pbxType <pbxType>]`

Description TeamCall Server starts up using the settings as specified in the main configuration file `<configFile>` as an absolute path (instead of using the configuration in the file `Default.conf`). If this file was not found, TeamCall Server will terminate with an error message.

Syntax `CSTAServer -a <cstaLink TCP/IP address> -c <configFile> [-pbxType <pbxType>]`

Description TeamCall Server receives the TCP/IP address of the phone system from the command line and all other settings from the specified main configuration file `<configFile>` as an absolute path. If the `<cstaLink TCP/IP Address>` is invalid, TeamCall Server will terminate with an error message.

Note: The following command line option is for the Linux version of TeamCall Server only.

Syntax `CSTAServer -d [-pbxType <pbxType>]`

Description Run TeamCall Server as a daemon. When using the option `-d` the `CSTAServer` process gets forked and sent into the background. This is especially useful when starting TeamCall Server automatically using a startup script.

Note: The following command line options are for the Windows version of TeamCall Server only.

Syntax `CSTAServer -i [-pbxType <pbxType>]`

Description Install as Windows Service option. This option creates an entry for TeamCall Server in the list of all processes to be started when Windows boots. This option cannot be combined with the following options: `-a`, `-c`, `-u` and `-s`. The main configuration file has to be placed in the same directory where the TeamCall Server binary is installed.

Syntax `CSTAServer -u [-pbxType <pbxType>]`

Description Uninstall as Windows Service option. This option removes the entry from the list of all processes to be started when Windows boots. This option cannot be combined with the following options: `-a`, `-c`, `-i` and `-s`.

Syntax `CSTAServer -s [-pbxType <pbxType>]`

Description Do not run as Windows Service option. When starting TeamCall Server from within the `Command Prompt` application on Windows, it will run as a background process. Starting it with the `-s` option runs TeamCall Server as a normal process. This option can be combined with the `-a` and `-c` options, but not with `-i` and `-u` options.

Example session: Starting TeamCall Server

For normal use (i.e., `debugLevel=0`) TeamCall Server should be run as a background process.

To start TeamCall Server on the command line, type:

```
CSTAServer -pbxType <pbxType>
```

under Windows or

```
CSTAServer -pbxType <pbxType> &
```

under Linux or macOS.

When starting, TeamCall Server returns status messages.

Stopping TeamCall Server

In order to stop the TeamCall Server service, a SuperVisor session must be established via the login port. Typing `SHUTDOWN` in the SuperVisor session will stop TeamCall Server. All sessions will display the message `SHUTDOWN`. In the example below, the login port is 26535.

Example

```
telnet localhost 26535
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

SuperVisor
error_ind SUCCESS SuperVisor

SHUTDOWN
error_ind SUCCESS SHUTDOWN
Connection closed by foreign host.
TeamCall Server has been shut down.
```

4

Administering TeamCall Server using a SuperVisor session

The SuperVisor interface allows to open a network connection to TeamCall Server via telnet and remotely administer settings of TeamCall Server.

For more information about CSTA specific error messages visit the web site of ECMA at www.ecma.ch. You will find a list of ilink specific error messages in the section *Ending the SuperVisor session* on page 30.

In order to open a connection to TeamCall Server, you'll have to know the server's application interface port number. This port number is defined in the main configuration file. In the example below the port number is 26535.

When a telnet connection has been established, a SuperVisor session can be started via the `SuperVisor` command.

The SuperVisor session allows an administrator to view the current configuration, enable/disable logging, change paths for logging, reload a new configuration from a specified configuration file (for example to change the debugging level) and manually add and/or remove devices to be monitored.

Example

Starting a SuperVisor session:

```
telnet 127.0.0.1 26535
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.

SuperVisor
error_ind SUCCESS SuperVisor
```

Viewing the current configuration

To view the current configuration, type `Config` in the SuperVisor session.

Example

Config

```
error_ind SUCCESS Config:
switchVendor : Alcatel
switchId : A4400
switchRelease : any
cstaPhase : 2
logDir : /opt/CSTAServer/etc/logs
binDir : /opt/CSTAServer/etc/
errorLogFile : Error.log
sysLogFile : Sys.log
debugLevel : 9
logFileMaxSize : 100
logFileMaxBackups : 30
moduleTimeout : 15
license :
1ac6w7af928aedr3ddd3f3920208167519720e641b9c31361ed647c23bf4f2b74ae
0612ca3609ab9ffdb58681650efa5a7fb88077d143bcf5e20899fe856c5d2064a93
1256b43f4c9922c05d0d3a8e4c1af34b1bb9f58bd6b17cf4cab423f879
deviceLogFile : Devices.log
callLogFile : Calls.log
cstaLogFile : CSTA.log
cstaLinkAddress : 10.0.0.1
cstaLinkPort : 2555
loginPort : 26535
adminAddress : 10.0.0.2
callLogEnabled : 0
cstaLogEnabled : 1
securityEnabled : 1
snapshotDevicePeriod : 300
maxPendingRequests : 20
requestTimeout : 30
linkNewA6 : 1
deviceMonitorConfFile : /opt/CSTAServer/config/DeviceMonitor.conf
prefixPlanAsString : +00/00:+0/0:+/00
prefixPlan : ((+00,00), (+0,0), (+,00))
```

TeamCall Server reports the current settings as listed above.

Activating the changes of a configuration file

To run TeamCall Server with new settings from a changed configuration file, TeamCall Server must be shut down and started up again.

Viewing license information

To view the license information, type `License` in the SuperVisor session.

Syntax

`License`

Description

Displays license information of TeamCall Server.

organization: Name of the organization, TeamCall Server is licensed to.

maxMonitors: Maximum number of different devices that can be monitored simultaneously among all sessions.

days left: Validity of the license in days.

expired: Shows if the license is expired: yes/no.

switch: type of the licensed phone system(s). Possible values are: "Any" or one or more values like "Hicom150", "A4400", "A4200", "A6", "HiPath8000"

Example

License

```
error_ind SUCCESS License
organization: MAX Communications, Inc.
maxMonitors: 150
days left: 316
expired: no
switch: Any
```

Viewing supported requests

In general, a phone system supports only some of the requests that are defined in CSTA Phase I, II or III. On top of that, TeamCall Server does not yet support all of the requests defined in CSTA Phase I, II or III. Therefore, TeamCall Server has a built-in list of supported and unsupported requests of each phone system specified using the key `switchVersion`.

Syntax

`IsSupported <Request>`

Description

Returns 1 (true) or 0 (false), if the request is supported by the phone system.

Examples

AnswerCall is supported:

IsSupported AnswerCall

```
error_ind SUCCESS IsSupported AnswerCall 1
```

SnapshotDevice is supported:

IsSupported SnapshotDevice

```
error_ind SUCCESS IsSupported SnapshotDevice 1
```

MakePredictiveCall is unsupported:

IsSupported MakePredictiveCall

```
error_ind SUCCESS IsSupported MakePredictiveCall 0
```

Syntax

`SupportedRequests`

Description

Returns a list with all supported requests. The number right before each request indicates that the request is enabled (1) or disabled (0).

Example

SupportedRequests

```
error_ind SUCCESS SupportedRequests
0 AlternateCall
1 AnswerCall
1 ClearConnection
1 ConferenceCall
1 ConsultationCall
0 DivertCallDeflect
0 HoldCall
1 MakeCall
1 QueryDevice
1 ReconnectCall
1 RetrieveCall
1 SetDeviceFeatureMessageWaiting
1 SetDeviceFeatureForwardImmediateOn
1 SetDeviceFeatureForwardImmediateOff
1 SetDeviceFeatureAgentLoggedIn
1 SetDeviceFeatureAgentLoggedOut
1 SetDeviceFeatureAgentNotReady
1 SetDeviceFeatureAgentReady
1 SetDeviceFeatureAgentBusy
1 SetDeviceFeatureAgentWorkingAfterCall
0 SingleStepTransfer
1 TransferCall
0 EscapeService
0 ChangeMonitorFilter
1 MonitorStart
1 MonitorStop
0 SnapshotDevice
```

Logging

Logging allows viewing a couple of different events. However, to enable logging, a valid path for creating logfiles must be specified. If no path has been set within the main configuration file or the path is invalid, log files will be created in the same folder where the TeamCall Server binary is stored.

Note: The size of the log files will only be limited by the available disk space of the volume, where the log files were created and the OS's file size restrictions. To ensure that logging will not stop the whole system by running out of space when logging a long period of time, the size of log files should be watched or a separate volume with limited size for logging should be used.

The log files

When logging has been activated, up to five different log files will be created depending on the chosen debug level:

An error log file named `Error.log`, a device log file named `Devices.log`, a calls log file named `Calls.log`, a system log file named `Sys.log` and a CSTA log file named `CSTA.log`.

The error log file contains everything that will be prompted to `stdout`.

The device log file contains information about the state of the monitored devices.

The calls log file contains information about the calls and the devices which are connected to these calls and their states.

The sys log file contains status, warning and error messages.

The CSTA log file contains the plain ASN1 Messages that were exchanged between TeamCall Server and the phone system.

The level of detail of the logged messages depends on the debug level. Refer to the upper sections for details about the debug levels.

Temporary changing path for log files

Typing `SetLogDir <Path>` temporarily changes the path for logging. `<Path>` must be a valid absolute path. After shutting down and starting up TeamCall Server, the log file path will be reset to the value of the main configuration file.

Example

```
SetLogDir /tmp/a4400
error_ind SUCCESS SetLogDir
```

Enable/disable logging

Logging of call events and CSTA events can be enabled and disabled by issuing the following requests:

Example

```
EnableLog call
error_ind SUCCESS EnableLog
DisableLog call
error_ind SUCCESS DisableLog
EnableLog csta
error_ind SUCCESS EnableLog
DisableLog csta
error_ind SUCCESS DisableLog
```

In contrast to the CSTA log and call log, which can be enabled and disabled, an error log file will always be generated if `logDir` exists. All messages from TeamCall Server which are sent to `stderr` will also be stored in this file.

Debugging modes

The standard debug level of TeamCall Server is 0. Only for troubleshooting purposes the debug level should be changed to a value different to 0. See chapter *Troubleshooting* on page 43.

Note: Debugging affects the performance of TeamCall Server.

The currently supported debug levels are as follows:

```
# 0: Enable logging for performance analysis
# 1: Allow logging of CSTA calls
# 2: Allow logging of devices (adding and removing)
# 5: ASN1 message decoding and logging (CSTAxAdaptor)
# 6: System status messages, e.g., "link down" (CSTAxAdaptor)
```

```
# 7: Messages between model and adaptor (CSTAController)
# 8: Protocol trace between adaptor and phone system
    (CSTAxAdaptor)
# 9: Messages between adaptor and phone system (CSTAxAdaptor)
```

Monitoring Devices

Adding/removing new devices to be monitored "stickily"

When starting TeamCall Server, a monitoring point in the phone system will be created for every device specified within the device configuration file. These devices will be monitored from startup to shut down of TeamCall Server. To add or remove devices to monitor until shut down of TeamCall Server without shutting down TeamCall Server, the requests:

```
AddDeviceMonitor <Device> and
```

```
RemoveDeviceMonitor <Device>
```

that must be issued in a SuperVisor session, allow to manually add or remove monitoring points. The monitoring points are "sticky", that means they persist until TeamCall Server is shut down.

Example

```
AddDeviceMonitor 123
```

```
## Add a Monitoring Point for Device 123
```

```
RemoveDeviceMonitor 456
```

```
## Remove Monitoring Point for device 456
```

Note: To successfully add a monitoring point for a device, the (internal) device number must be valid. Only one monitoring point per device can be set. To successfully remove a monitoring point for a device, the device number must be valid and there must be an existing monitoring point for it. Otherwise an error will be reported.

These settings are temporary settings. After shutting down and starting up TeamCall Server, the monitoring points within the phone system will be reset according to the values loaded from the device configuration file.

More about device monitoring

Device monitoring allows you to gather continuous information about device changes. To view these changes, a `MonitorStart` request must be issued in an STLI session (see *MonitorStart* in the STLI Reference Guide). For important additional information, refer to *Important notice* on page 18.

As mentioned in the previous sections, there are three levels of device monitoring which affect the period of time TeamCall Server monitors a device.

There are three different ways to monitor a device:

- permanently by adding this device to the device configuration file
- "sticky" by calling an `AddDeviceMonitor` request in a SuperVisor session
- dynamically by only calling a `MonitorStart/TSPI_lineOpen` request in an STLI session.

If there is no entry in the device configuration file and no device was added using a SuperVisor session, TeamCall Server only monitors the devices which are dynamically introduced using the `MonitorStart/TSPI_lineOpen` request. Use the `MonitorStop/TSPI_lineClose` request to remove the dynamically introduced monitoring points.

The second way to configure monitoring points within the TeamCall Server configuration is to add the devices to be monitored to the device configuration file.

At last, monitoring points can also be added in a "sticky" way using the `AddDeviceMonitor` request within a SuperVisor session.

The difference between these two ways is that if a device has been added to the device configuration file, TeamCall Server internally traces change information about this device from startup to shut down.

Also, the specified device will be monitored every time TeamCall Server starts up, while a device which is added using the `AddDeviceMonitor` request will only be monitored as long as TeamCall Server is running.

Hint: Add a new device using the request `AddDeviceMonitor` and additionally add it to the device configuration file (also see *Important notice* on page 18) to avoid restarting TeamCall Server.

Syntax `ShowDeviceMonitors`

Description Lists all currently monitored devices in the following format: `<device> <isSticky> <isActive> <number of observers>`. For more information about the flag `isSticky` see *Monitoring Devices* on page 28. The flag `isActive` shows if the monitoring request has already been answered by the phone system. The last parameter shows how many observers (i.e., sessions) are currently monitoring the device.

Example `ShowDeviceMonitors`

```
error_ind SUCCESS ShowDeviceMonitors
111 sticky:true active:true observers:11
101 sticky:true active:true observers:25
168 sticky:false active:true observers:3
185 sticky:true active:true observers:9
123 sticky:true active:false observers:0
187 sticky:true active:true observers:9
172 sticky:true active:true observers:0
173 sticky:false active:true observers:1
```

Note: Any attempt to start monitoring on a non-active DeviceMonitor will be answered with `error_ind PENDINGREQUEST <Request>`.

Syntax `MonitorObservers <device number>`

Description Lists details about all sessions, observing (monitoring) a device.

Output

```
error_ind SUCCESS MonitorObservers <number of observers>
ref:<internal reference to the observer> host:<IP address>
port:<number> type:<TAPISession or STLISession>
```

Example `MonitorObservers 111`

```
error_ind SUCCESS MonitorObservers 2
ref:142138112 host:10.10.6.21 port:4991 type:TAPISession
ref:141999616 host:10.10.7.15 port:1777 type:STLISession
```

Ending the SuperVisor session

The BYE command can be used to end the SuperVisor session and disconnect from the server.

Example **BYE**
 error_ind SUCCESS BYE

Error messages

When issuing requests, errors can occur. The following list describes all TeamCall Server specific error messages. These are written in UPPERCASE. All other error messages appearing in mixed case, like "UniversalFailure" are CSTA specific. For more information about CSTA specific error messages visit the web site of ECMA at www.ecma.ch.

FILEERROR

Response error_ind FILEERROR SetLogDir
Description Directory does not exist or is read only.
Example **SetLogDir /doesNotExist**
 error_ind FILEERROR SetLogDir

ILINK-INTERNAL DEMOVERSION

Response error_ind ILINK-INTERNAL DEMOVERSION MonitoringPointsExceeded
Description Only one device can be monitored with this demo version.

INTERNALERROR

Response error_ind INTERNALERROR <Request>
Description Internal error. Please analyze the Error.log file.

INVALCMD

Response error_ind INVALCMD <Request>
Description Unknown request.
Example **Add 123**
 error_ind INVALCMD Add

INVALNUMPARAM

Response error_ind INVALNUMPARAM <Request>
Description Invalid number of parameters for this request.
Example **AddDeviceMonitor 524 123**
 error_ind INVALNUMPARAM AddDeviceMonitor

INVALPARAM

Response error_ind INVALPARAM <Request>

Description Wrong parameter(s) for this request.

Example EnableLog ddd
error_ind INVALPARAM EnableLog

LICENSEERROR

Response error_ind LICENSEERROR <message>

Description Either the request cannot be executed because of license limitations (e.g., demo version) or the license has expired.

Examples MonitorStart 112
error_ind LICENSEERROR MONITORINGPOINTSEXCEEDED Monitor not set
MonitorStart 112
error_ind LICENSEERROR LICENSEEXPIRED License expired since <x>
days

LINKOUTOFSERVICE

Response error_ind LINKOUTOFSERVICE <Request>

Description The request cannot be executed because there is no connection to the phone system at this moment. When the connection to the phone system is working again, the message LINKBACKINSERVICE" will be shown.

Example AddDeviceMonitor 524
error_ind LINKOUTOFSERVICE AddDeviceMonitor

NOMULTIPLEINSTANCE

Response error_ind NOMULTIPLEINSTANCE AddDeviceMonitor

Description The request AddDeviceMonitor has already been issued for this device during this session.

Example AddDeviceMonitor 524
error_ind SUCCESS AddDeviceMonitor
AddDeviceMonitor 524
error_ind NOMULTIPLEINSTANCE AddDeviceMonitor

NOTREGISTERED

Response error_ind NOTREGISTERED RemoveDeviceMonitor

Description The device is not (or no longer) monitored during this session.

Example RemoveDeviceMonitor 524
error_ind SUCCESS RemoveDeviceMonitor
RemoveDeviceMonitor 524
error_ind NOTREGISTERED RemoveDeviceMonitor

PENDINGREQUEST

Response `error_ind PENDINGREQUEST <Request>`

Description Another session is currently waiting for the same request with the same parameters. Usually, this error is sent when another session already started a monitoring request for the same device and the request has not yet been answered by the phone system (i.e., the DeviceMonitor is *not* active).

Example `AddDeviceMonitor 111`
`error_ind PENDINGREQUEST AddDeviceMonitor 111`

SUCCESS

Response `error_ind SUCCESS <Request>`

Description Not an error at all.

Example `AddDeviceMonitor 524`
`error_ind SUCCESS AddDeviceMonitor`

UNAVAILBLEREQUEST

Response `error_ind UNAVAILBLEREQUEST <request> ""`

Description The issued request is not supported. It could be not supported at all, disabled or invalid.

5

APIs

TeamCall CSTA Server offers a number of APIs that can be used to implement CTI applications that connect to different phone systems:

CSTA

CSTA (Computer Supported Telecommunication Applications) is an industry standard, described by ECMA, an international Europe-based industry association founded in 1961 and dedicated to the standardization of information and communication systems.

For more information visit their web site at www.ecma.ch.

There are ten CSTA standards in total, you may want to start with these:

- ECMA-269: Services for Computer Supported Telecommunications Applications (CSTA) Phase III
- ECMA-285: ASN.1 for Computer Supported Telecommunications Applications (CSTA) phase III
- ECMA-323: XML protocol for Computer Supported Telecommunications Applications (CSTA) Phase III

TeamCall CSTA Server offers native CSTA support for applications.

JTAPI

The Java Telephony API (JTAPI) is Java's "official" API for CTI / Call control.

See www.oracle.com/java/technologies/jtapi.html

TeamCall CSTA Server offers JTAPI support for applications via its native JTAPI Java library.

STLI

STLI (Simple Telephony Interface) is a human readable plaintext CTI API based on CSTA.

For more information see the separate *STLI Reference Guide*.

TeamCall CSTA Server offers native STLI support for applications.

TAPI

TAPI (Telephony Application Programming Interface) is Windows' "official" CTI API. TeamCall CSTA Server offers TAPI support for applications via its native Windows TSP (TAPI Service Provider).

6

Log files

TeamCall Server logs all of its activities in different log files when activated and configured within the main configuration file. You can turn on the logging mechanism to debug an error, but remember to turn this feature off, when it is no longer needed.

The number of messages written to the log files depend on the debug level set within the main configuration file. To learn more about debug levels, see the configuration key `debugLevel` on page 11.

The log files have the following names:

Error.log – The logging mechanism does not need to be turned on as described above for the error logging feature. The error log file is created automatically when the server starts up.

Sys.log – Status messages, warnings and errors. For detailed information see *Sys.log* on page 36.

Device.log – It contains information about the status of the monitored telephones (devices).

Calls.log – The `Calls.log` file contains information about the calls and the telephones that are connected to these calls and their status.

CSTA.log – The `CSTA.log` file contains the plain ASN.1 messages that are exchanged between TeamCall Server and the phone system.

Activate logging

Note: Activating the logging mechanism may affect the performance of the server.

To enable the logging mechanism, follow these steps:

1. Locate the main configuration file (e.g., `Default.conf`).
On a computer running Windows, the file `Default.conf` is located in the installation path.
2. Launch your favorite text editing application.
3. Open the file `Default.conf`.
4. Find the entry `callLogEnabled`.
5. Change the value after `callLogEnabled` to 1 in order to activate call logging.
6. Find the entry `cstaLogEnabled`.

7. Change the value after `cstaLogEnabled` to 1 in order to activate CSTA logging.
8. Find the entry `debugLevel`.
9. Change the value after `debugLevel` to 9 in order to enable full logging.
10. Find the entry `logFileMaxSize`.
11. Change the value after `logFileMaxSize` to -1 in order to disable log rotation.

Note: If any of these entries are not present in the file, please add them.

Sys.log

The following messages are logged in the file `Sys.log` in the log directory. There are three categories for messages:

- Status – just for your information
- Warnings – notes about problems that can occur under certain conditions
- Errors – errors that **have** to be corrected in order to run TeamCall Server correctly.

Information in `<>` represent variables.

Status

```
Connected to Admin Server.
Connected to PBX/Link, sending login sequence.
Connecting to <value>:<value> (IP <value>)...
Connecting to Admin Server at <aaa.bbb.ccc.ddd:port>
Connection terminated.
Connection to Admin Server terminated.
Log in to PBX/Link successful.
PBX Link up
Reloading DeviceMonitors (PBX LinkUp)
STLI port <port> added.
Server shutdown.
SuperVisor port <port> added.
Waiting for connection to PBX/Link.
```

Warnings (WRNXXXX)

WRN0003

- Message** (WRN0003) DEMOVERSION restricted to one monitoring point.
- Description** You are currently using a demo version of TeamCall Server which is restricted to one monitoring point only.

WRN0004

- Message** (WRN0004) SuperVisor port <port> not available, retrying.
- Description** The SuperVisor port entered in the main configuration file (key: servicePort, default: 7999) is currently not available. TeamCall Server will retry to connect to the port every 2 minutes. If the port is still not available after 5 retries you should either use another port or check if there is not another instance of TeamCall Server running on the same computer using the same port.

WRN0005

- Message** (WRN0005) STLI port <port> not available, retrying.
- Description** The STLI port entered in the main configuration file (key: stliPort, default: 8000) is currently not available. TeamCall Server will retry to connect to the port every 2 minutes. If the port is still not available after 5 retries you should either use another port or check if there is not another instance of TeamCall Server running on the same computer using the same port.

WRN0006

- Message** (WRN0006) NetTSPI port %d not available, retrying.
- Description** The NetTSPI port entered in the main configuration file (key: netTspiPort, default: 26535) is currently not available. TeamCall Server will retry to connect to the port every 2 minutes. If the port is still not available after 5 retries you should either use another port or check if there is not another instance of TeamCall Server running on the same computer using the same port.

WRN0007

- Message** (WRN0007) DEMOVERSION restricted to one monitoring point: <value> not set
- Description** You are currently using a demo version of TeamCall Server which is restricted to one monitoring point only.

WRN0009

- Message** (WRN0009) Error Processing line <value> in file <value>: <value>.
- Description** An entry in the main configuration file could not be processed. The entry will be ignored. Please check the main configuration file.

WRN0010

Message (WRN0010) SupportedRequest file <value> : invalid entry "<value>", ignoring

Description An entry in your SupportedRequest file could not be processed. The entry has been ignored. Please check your SupportedRequest file if you experience problems.

WRN0011

Message (WRN0011) SupportedRequest file <value> : invalid/unsupported request in entry ``<value>'', ignoring

Description An entry in your SupportedRequest file could not be processed. The entry has been ignored. Please check your SupportedRequest file if you experience problems.

WRN0012

Message (WRN0012) SupportedRequest file <value> : invalid value in entry "<value>", ignoring

Description An entry in your SupportedRequest file could not be processed. The entry has been ignored. Please check your SupportedRequest file if you experience problems.

WRN0013

Message (WRN0013) Login port <value> not available, retrying.

Description The login port given in the main configuration file (key: loginPort, default: 26535) is currently not available. TeamCall Server will retry to connect to the port every 2 minutes. If the port is still not available after 5 retries you should either use another port or check if there is not another instance of TeamCall Server running on the same computer using the same port.

Errors (ERRXXXX)

ERR0001

Message (ERR0001) Connection to TeamCall Admin Server failed (err = <value>)

Description Connection to TeamCall Admin Server failed. Please check the given IP address in the main configuration file of TeamCall Server (key: adminAddress, default: 127.0.0.1) and the installation of the TeamCall Admin Server.

ERR0002

Message (ERR0002) Hostname for administration (<value>) could not be resolved to IP address, check your DNS.

Description The DNS name for the TeamCall Admin Server set in the main configuration file (key: `adminAddress`) is not correct: the specified hostname cannot not be mapped to a TCP/IP address. Please check the entry in the main configuration file and also the DNS configuration of the TeamCall Admin Server computer.

ERR0003

Message (ERR0003) PBX link down

Description TeamCall Server has currently no connection to the phone system. Please check the network connectivity of the host where TeamCall Server is running. Also make sure that the phone system is still connected to the local network and that it is up und running.

ERR0004

Message (ERR0004) PBX link idle

Description TeamCall Server has currently no connection to the phone system. Please check the network connectivity of the host where TeamCall Server is running. Also make sure that the phone system is still connected to the local network and that it is up und running.

ERR0005

Message (ERR0005) Session login failed

Description The entered password and/or userid is invalid. Please check your input and try again.

ERR0006

Message (ERR0006) Session login failed password expired

Description The entered password for that userId is no longer valid. Please check your input and try again.

ERR0009

Message (ERR0009) Can't open device monitor configuration file <value>

Description The value of the key `deviceMonitorConfFile` in the main configuration file is not correct: the file could not be found. Please enter the complete path of the file and restart TeamCall Server.

ERR0010

- Message** (ERR0010) `setsockopt for port <value> failed <value>: <value>`
- Description** The port is currently not useable. Please locate the port, edit the main configuration file, choose a different port number and restart TeamCall Server. If you still experience this problem, please enable debugging, restart the server, replay the same scenario and send the log files to the TeamCall support department.

ERR0011

- Message** (ERR0011) `Can't open stream socket for port <value>`
- Description** The port is currently not useable. Please locate the port, edit the main configuration file, choose a different port number and restart TeamCall Server. If you still experience this problem, please enable debugging, restart the server, replay the same scenario and send the log files to the TeamCall support department.

ERR0012

- Message** (ERR0012) `Hostname <value> could not be resolved to IP address, check your DNS`
- Description** The DNS name for the phone system entered in the main configuration file of TeamCall Server (key: `cstaLinkAddress`) is not set correctly. The specified host name cannot be mapped to a TCP/IP address. Please check the entry in the main configuration file and also the DNS configuration of the TeamCall Server computer.

ERR0013

- Message** (ERR0013) `Connection to PBX/Link failed (err = <value>)`
- Description** TeamCall Server will retry to connect to the phone system every 2 minutes. If the connection still fails after 5 retries, you should make sure that:
- the key `cstaLinkAddress` in the main configuration file is set correctly
 - the key `cstaLinkPort` in the main configuration file is set correctly
 - the TeamCall Server computer is connected to the network
 - the phone system is available within the network
 - the phone system is up and running
 - the TeamCall Link process is up and running
 - there is no other TeamCall Server currently connected to the phone system

ERR0014

- Message** (ERR0014) `PBX unavailable: received "U"`
- Description** This error is Alcatel specific. The Alcatel phone system is currently not available. TeamCall Server will retry to connect to the phone system every 2 minutes. If the connection still fails after 5 retries, you should check your phone system.

ERR0015

Message (ERR0015) Signal caught: timeout <signal>

Description TeamCall Server received a signal and terminated. Please enable debugging, restart the server, replay the same scenario and send the log files to the TeamCall support department.

ERR0016

Message (ERR0016) Signal caught: restart <signal>

Description The server received a signal and terminated. Please enable debugging, restart the server, replay the same scenario and send the log files to the TeamCall support department.

ERR0017

Message (ERR0017) Signal caught: alarm_handler <signal>

Description TeamCall Server received a signal and terminated. Please enable debugging, restart the server, replay the same scenario and send the log files to the TeamCallsupport department.

ERR0018

Message (ERR0018) Signal caught: just_die <signal>

Description The server received a signal and terminated. Please enable debugging, restart the server, replay the same scenario and send the log files to the TeamCall support department.

ERR0019

Message (ERR0019) Could not write login message

Description This error is Hicom150 specific. The initial login message could not be sent to the Hicom150. Please check to make sure that:

- the key `cstaLinkAddress` in the main configuration file is set correctly
- the key `cstaLinkPort` in the main configuration file is set correctly
- the phone system and Link is available within the network
- the TeamCall Server computer is connected to the network
- the phone system and TeamCall Link is up and running
- there is no other TeamCall Server currently connected to the TeamCall Link

ERR0020

Message (ERR0020) Log in to PBX/Link failed: received LOGNACK

Description This error is Hicom150 specific. Apparently, someone modified the password of the Hicom150. Either reset it to the original password or contact the TeamCall support department in order to get a modified version of TeamCall Server.

ERR0026

Message (ERR0026) Missing mandatory key "<key>" in Configuration File!Killing Process.

Description The mandatory <key> has to be set in the main configuration file.

ERR0030

Message (ERR0030) start: Can't open stream socket.

Description TeamCall Server cannot connect to the phone system and will retry to connect to the phone system every 2 minutes. If the connection still fails after 5 retries you should make sure that:

- the key `cstaLinkAddress` in the main configuration file is set correctly
- the key `cstaLinkPort` in the main configuration file is set correctly
- the phone system is available within the network
- the TeamCall Server computer is connected to the network
- the phone system is up and running
- the TeamCall Link process is up and running
- there is no other TeamCall Server currently connected to the phone system

ERR0031

Message (ERR0031) Can't open supportedRequest file %s <value>

Description The value of the key `supportedRequest` in the main configuration file is not set correctly. The file could not be found. Please enter the file name with the absolute path and restart TeamCall Server.

7

Troubleshooting

In this chapter you find information about known issues with TeamCall Server. Not all of these issues apply to the current release. Issues that are no longer present in the current release may be of interest to developers who develop software based on the current release but plan to deploy it on a previous release.

Issues

After starting TeamCall Server, no connection to the phone system is established

Operating systems: All

Description TeamCall Server has been started. The current configuration has been reported. Below this configuration listing, TeamCall Server reports

```
Connecting to 195.21.11.162:2555 ...
```

```
Waiting for connection to server.
```

(where 195.21.11.162:2555 is TCP/IP:PORT in this example) will be reported but no connection can be established

Work-around

The network connection to the phone system cannot be established. Check out the network in the following order:

Try to ping the TCP/IP address of the phone system. If all packets get lost, the TCP/IP address may be wrong, the physical network connections may not work correctly or the phone system may have no network port.

If pinging was successful, try a telnet to TCP/IP:PORT of the phone system. If it is refused, and you get an error message like

```
telnet: Unable to connect to remote host: Connection refused
```

the TCP/IP port number might be wrong; check the TCP/IP port number of the phone system. Another indicator for a wrong port number at the phone system is that additionally to the waiting to connect message, a message like

```
Connection terminated.
```

will be reported.

If both previous tests were successful, it is possible that TCP/IP:PORT is valid, but does not belong to the phone system.

In general, check the network configuration.

Local devices cause an error, when accessed with STLI requests

Operating systems: All

Description After sending some requests (e.g., `MakeCall` or `ConsultationCall`) TeamCall Server responds with

```
error_ind UniversalFailure operationalErrors  
requestIncompatibleWithObject
```

or after sending other requests (e.g., `AnswerCall`) TeamCall Server responds with

```
Failed <monitoredDevice> destNotObtainable <failedDevice>  
<calledDevice>
```

Work-around

There are two possible reasons:

1.

If a device is handled by TeamCall Server in a phone system installation, it has to be announced to TeamCall Server/phone system by setting a monitoring point for it.

This can be done by adding its device number to the device configuration file. For temporary cases and test cases it can be added manually in SuperVisor mode, calling the `AddDeviceMonitor` request.

In the described error case, the first possible reason is, that there is no monitoring point for the device. A device without a monitor point can be accessed by a `MakeCall` request, but not by an `AnswerCall` request.

2.

The second possible reason is that the device has been physically disconnected (unplugged) or is in any other way not correctly connected to the phone system or setup at phone system.

First make sure that the device is connected to the phone system and that it works correctly in this environment, by taking it off-hook and manually calling another device.

If there is no dial tone or another device cannot be called manually, check the phone system and its configuration.

If the previous test was successful, the device cannot be monitored. Check the configuration of the phone system.