

# CSTA

---

CSTADLL  
Version 2.0.4  
CSTADLL  
Reference Manual



The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of this agreement.

### **Copyright Notice**

Copyright ©1997-2013 Objective Systems, Inc. All rights reserved.

This document may be distributed in any form, electronic or otherwise, provided that it is distributed in its entirety and that the copyright and this notice are included.

### **Author's Contact Information**

Comments, suggestions, and inquiries regarding CSTADLL may be submitted via electronic mail to [info@obj-sys.com](mailto:info@obj-sys.com).



# Contents

<b>1</b>	<b>CSTADLL</b>	<b>1</b>
<b>2</b>	<b>Namespace Documentation</b>	<b>3</b>
2.1	Package Com.Objsys.Csta.Common . . . . .	3
2.1.1	Detailed Description . . . . .	3
2.2	Package Com.Objsys.Csta.Devices . . . . .	4
2.2.1	Detailed Description . . . . .	4
2.3	Package Com.Objsys.Csta.Phase1 . . . . .	5
2.3.1	Detailed Description . . . . .	5
2.4	Package Com.Objsys.Csta.Phase2 . . . . .	6
2.4.1	Detailed Description . . . . .	6
2.5	Package Com.Objsys.Csta.Phase3 . . . . .	7
2.5.1	Detailed Description . . . . .	7
<b>3</b>	<b>Class Documentation</b>	<b>9</b>
3.1	Alcatel4400 Class Reference . . . . .	9
3.1.1	Detailed Description . . . . .	9
3.1.2	Constructor & Destructor Documentation . . . . .	9
3.1.2.1	Alcatel4400 . . . . .	9
3.1.3	Member Function Documentation . . . . .	9
3.1.3.1	MakeACSEAssociation . . . . .	9
3.2	AlcatelOXO Class Reference . . . . .	10
3.2.1	Detailed Description . . . . .	10
3.2.2	Constructor & Destructor Documentation . . . . .	10
3.2.2.1	AlcatelOXO . . . . .	10
3.2.3	Member Function Documentation . . . . .	10
3.2.3.1	MakeACSEAssociation . . . . .	10
3.3	AlcatelOXOp1 Class Reference . . . . .	11
3.3.1	Detailed Description . . . . .	11

3.3.2	Constructor & Destructor Documentation	11
3.3.2.1	AlcatelOXOp1	11
3.3.3	Member Function Documentation	11
3.3.3.1	MakeACSEAssociation	11
3.4	CSTAContext Class Reference	12
3.4.1	Detailed Description	12
3.4.2	Property Documentation	12
3.4.2.1	ResponseFromPBX	12
3.4.2.2	ResponsesFromPBX	12
3.5	CSTAResponseInfo Class Reference	13
3.5.1	Detailed Description	13
3.5.2	Property Documentation	13
3.5.2.1	ResponseFromPBX	13
3.5.2.2	ResponsesFromPBX	13
3.5.2.3	StatusCode	13
3.5.2.4	StatusMessage	13
3.6	GenericCSTAp1 Class Reference	14
3.6.1	Detailed Description	14
3.6.2	Constructor & Destructor Documentation	14
3.6.2.1	GenericCSTAp1	14
3.6.3	Member Function Documentation	14
3.6.3.1	AnswerCall	14
3.6.3.2	AnswerCall	15
3.6.3.3	AnswerCall	15
3.6.3.4	ClearConnection	15
3.6.3.5	ConsultationCall	15
3.6.3.6	DivertCall	16
3.6.3.7	EncodeROSERequestHeader	16
3.6.3.8	MakeACSEAssociation	16
3.6.3.9	MakeCall	16
3.6.3.10	MonitorStart	17
3.6.3.11	MonitorStop	17
3.6.3.12	MonitorStop	17
3.6.3.13	MonitorStop	17
3.6.3.14	QueryDevice	18
3.6.3.15	ReleaseACSEAssociation	18
3.6.3.16	SnapshotDevice	18

3.6.3.17	TransferCall	18
3.6.3.18	TransferCall	18
3.6.4	Property Documentation	19
3.6.4.1	ThreadContext	19
3.7	GenericCSTAp2 Class Reference	20
3.7.1	Detailed Description	20
3.7.2	Constructor & Destructor Documentation	20
3.7.2.1	GenericCSTAp2	20
3.7.3	Member Function Documentation	20
3.7.3.1	AnswerCall	20
3.7.3.2	AnswerCall	21
3.7.3.3	AnswerCall	21
3.7.3.4	ClearConnection	21
3.7.3.5	ConsultationCall	21
3.7.3.6	DivertCall	22
3.7.3.7	EncodeROSERequestHeader	22
3.7.3.8	MakeACSEAssociation	22
3.7.3.9	MakeCall	22
3.7.3.10	MonitorStart	23
3.7.3.11	MonitorStop	23
3.7.3.12	MonitorStop	23
3.7.3.13	MonitorStop	23
3.7.3.14	QueryDevice	24
3.7.3.15	ReleaseACSEAssociation	24
3.7.3.16	SnapshotDevice	24
3.7.3.17	TransferCall	24
3.7.3.18	TransferCall	24
3.7.4	Property Documentation	25
3.7.4.1	ThreadContext	25
3.8	GenericCSTAp3 Class Reference	26
3.8.1	Detailed Description	26
3.8.2	Constructor & Destructor Documentation	26
3.8.2.1	GenericCSTAp3	26
3.8.3	Member Function Documentation	27
3.8.3.1	AnswerCall	27
3.8.3.2	AnswerCall	27
3.8.3.3	AnswerCall	27

3.8.3.4	ClearConnection	27
3.8.3.5	ClearMessageWaiting	28
3.8.3.6	ConsultationCall	28
3.8.3.7	EncodeROSERequestHeader	28
3.8.3.8	GetSFDevices	28
3.8.3.9	GetSFDevices	29
3.8.3.10	MakeACSEAssociation	29
3.8.3.11	MakeCall	29
3.8.3.12	MonitorStart	29
3.8.3.13	MonitorStop	29
3.8.3.14	MonitorStop	30
3.8.3.15	MonitorStop	30
3.8.3.16	ReleaseACSEAssociation	30
3.8.3.17	RingDevice	30
3.8.3.18	SendData	31
3.8.3.19	SetDisplay	31
3.8.3.20	SetMessageWaiting	31
3.8.3.21	SingleStepTransfer	31
3.8.3.22	SingleStepTransfer	32
3.8.3.23	SnapshotDevice	32
3.8.3.24	StartDataPath	32
3.8.3.25	StopDataPath	32
3.8.3.26	StopRing	33
3.8.3.27	TransferCall	33
3.8.3.28	TransferCall	33
3.8.4	Property Documentation	33
3.8.4.1	ThreadContext	33
3.9	IETF_CSTAp1 Class Reference	34
3.9.1	Detailed Description	34
3.9.2	Constructor & Destructor Documentation	34
3.9.2.1	IETF_CSTAp1	34
3.10	IETF_CSTAp2 Class Reference	35
3.10.1	Detailed Description	35
3.10.2	Constructor & Destructor Documentation	35
3.10.2.1	IETF_CSTAp2	35
3.11	IETF_CSTAp3 Class Reference	36
3.11.1	Detailed Description	36



3.11.2	Constructor & Destructor Documentation	36
3.11.2.1	IETF_CSTAp3	36
3.12	LicenseException Class Reference	37
3.12.1	Detailed Description	37
3.13	PanasonicKXTDA Class Reference	38
3.13.1	Detailed Description	38
3.14	PanasonicKXTDE Class Reference	39
3.14.1	Detailed Description	39
3.14.2	Member Function Documentation	39
3.14.2.1	AcquireControlRight	39
3.14.2.2	ClearMessageWaiting	39
3.14.2.3	GetGroupMembers	40
3.14.2.4	GetSFDevices	40
3.14.2.5	PDFStart	40
3.14.2.6	PDFStop	40
3.14.2.7	ReleaseControlRight	41
3.14.2.8	ResetDisplay	41
3.14.2.9	SendKmeMessage	41
3.14.2.10	SetMessageWaiting	41
3.15	PanasonicNCP Class Reference	42
3.15.1	Detailed Description	42
3.16	PBXSessionException Class Reference	43
3.16.1	Detailed Description	43
3.17	PBXSessionHelper Class Reference	44
3.17.1	Detailed Description	44
3.17.2	Member Function Documentation	44
3.17.2.1	AsyncCallback	44
3.17.2.2	Close	44
3.17.2.3	ConnectionCallback	45
3.17.2.4	Init	45
3.17.2.5	Open	45
3.17.2.6	SendACSEMessage	45
3.17.2.7	SendMessage	45
3.17.2.8	SendMessage	46
3.17.2.9	WaitForROSEResponse	46
3.17.3	Property Documentation	46
3.17.3.1	ClientCallback	46

3.17.3.2	ConnectionLostCallback	46
3.17.3.3	DebugMode	46
3.17.3.4	LoggingEnabled	46
3.17.3.5	LoggingFolder	47
3.18	Phase1Opcodes Class Reference	48
3.18.1	Detailed Description	48
3.18.2	Member Enumeration Documentation	48
3.18.2.1	Opcodes	48
3.19	Phase2Opcodes Class Reference	49
3.19.1	Detailed Description	49
3.19.2	Member Enumeration Documentation	49
3.19.2.1	Opcodes	49
3.20	Phase3Opcodes Class Reference	50
3.20.1	Detailed Description	50
3.20.2	Member Enumeration Documentation	50
3.20.2.1	Opcodes	50
3.21	ROSEParseInfo Class Reference	51
3.21.1	Detailed Description	51
3.22	SiemensCap Class Reference	52
3.22.1	Detailed Description	52
3.23	SiemensHicom300 Class Reference	53
3.23.1	Detailed Description	53
3.23.2	Constructor & Destructor Documentation	53
3.23.2.1	SiemensHicom300	53
3.24	SiemensHipath3000p2 Class Reference	54
3.24.1	Detailed Description	54
3.24.2	Constructor & Destructor Documentation	54
3.24.2.1	SiemensHipath3000p2	54
3.25	SiemensHipath3000p3 Class Reference	55
3.25.1	Detailed Description	55
3.25.2	Constructor & Destructor Documentation	55
3.25.2.1	SiemensHipath3000p3	55
3.26	SiemensHipath4000 Class Reference	56
3.26.1	Detailed Description	56
3.27	SiemensRealitis Class Reference	57
3.27.1	Detailed Description	57
3.28	SingleStepTransferInfo Class Reference	58

3.28.1	Detailed Description	58
3.28.2	Property Documentation	58
3.28.2.1	TransferFromDevice	58
3.28.2.2	TransferringCallID	58
3.28.2.3	TransferToDevice	58
3.29	SocketState Class Reference	59
3.29.1	Detailed Description	59
3.29.2	Property Documentation	59
3.29.2.1	AckBuffer	59
3.29.2.2	ReadBuffer	59
3.29.2.3	ReadBuffers	59
3.29.2.4	TotalLength	59
3.30	TadiranCoral Class Reference	60
3.30.1	Detailed Description	60
3.31	TransferCallInfo Class Reference	61
3.31.1	Detailed Description	61
3.31.2	Property Documentation	61
3.31.2.1	ActiveDeviceCallID	61
3.31.2.2	ActiveDeviceNumber	61
3.31.2.3	ConnectedDevice	61
3.31.2.4	HeldDevice	61
3.31.2.5	HeldDeviceCallID	61
3.31.2.6	HeldDeviceNumber	61
3.32	UnifyOpenscapeX5 Class Reference	62
3.32.1	Detailed Description	62
3.32.2	Constructor & Destructor Documentation	62
3.32.2.1	UnifyOpenscapeX5	62



# Chapter 1

## CSTADLL

The CSTADLL product is a Microsoft .NET 3.5 DLL that allows client code to use a single library to communicate with a PBX device, regardless of what CSTA phase that device uses. A version of the DLL built with .NET 2.0 is also provided in the kit.

The DLL uses the following namespaces:

- `Com.Objsys.Csta.Common`
- `Com.Objsys.Csta.Devices`
- `Com.Objsys.Csta.Phase1`
- `Com.Objsys.Csta.Phase2`
- `Com.Objsys.Csta.Phase3`

The `Com.Objsys.Csta.Common` namespace contains classes that are common to all CSTA phases.

The `Com.Objsys.Csta.Devices` namespace contains classes that allow a caller to use specific PBX devices.

The `Com.Objsys.Csta.Phase(n)` namespaces contain classes that are specific to the indicated phase. Most of these classes are generated by ASN1C from the CSTA and ACSE ASN.1 specifications. These generated classes are not documented here, but you can consult the ASN1C C# User Guide for information about how ASN.1 constructions are translated into C# classes.

Each namespace also contains several classes that are not generated by ASN1C. These classes are the ones documented in this manual.

The DLL allows a client to have a session with a single PBX system, during which the client can send ACSE and CSTA messages to the PBX and receive responses.

The typical way to use the DLL is to use the `PBXSessionHelper` class to set up the communication to the PBX system via the `Init` method. If the PBX will be sending asynchronous data, such as monitor packets, to the client, the `ClientCallback` property can be used to define a callback method to receive the asynchronous data. If no callback method is defined, asynchronous data will be ignored.

The CSTADLL kit includes several samples to guide you in writing your own code. The name of the sample conveys some information about the sample. If the name of the sample starts with DLL, it means the sample shows how to use one of the helper methods that the DLL exposes. If the name of the sample starts with CSTA, then that sample still shows a way to make use of the DLL, but instead of using one of the DLL's helper methods, the sample instead shows how to use the DLL to send to a PBX a CSTA message for which the DLL doesn't expose a helper method.

After either DLL or CSTA in the sample name is an indicator of what language the sample is written in. Cs is used to indicate that the sample is written in C#. Vb will be used to indicate that the sample is written in Visual BASIC.

Similar codes will be used if samples are provided in other languages; e.g., perhaps CppCLI for samples written in C++/CLI.

For example, the sample `DLLCsAnswerCall` is a sample that shows how to use one of the DLL's helper methods in C# to instruct a PBX to answer a call. The sample `CSTACsButtonPress` shows how to use the DLL with C# to send a PBX the `CSTA ButtonPress` message, for which there is no exposed helper method.

The classes and methods exposed by the all-phases DLL within the `CSTADLL` package are probably sufficient to handle ACSE and CSTA operations for most PBX devices. But if needed, you can write a class of your own to handle ACSE and CSTA operations for a PBX device that the `CSTADLL` software doesn't explicitly support. The sample `DLLCsNewPBX` shows how this might be accomplished. This sample contains code for a small separate DLL that could be used to support a fictitious PBX device called the `AwesomePBX100`. The assumption in the sample is that this device uses standard phase 2 messages for all operations except for the ACSE make association message. This message is the one message that is most commonly different from one PBX to the next. The `DLLCsNewPBX` sample shows how the `MakeACSEAssociation` method within the `GenericCSTAp2` class can be overridden in a class that you can write. The override implementation handles the details that are specific to the device. Other methods within `GenericCSTAp2` could also be overridden as needed.

The all-phases DLL can log message traffic between a client program and the PBX device if so desired. The logging is controlled by the `LoggingEnabled` property with the `PBXSessionHelper` class. The logging is off by default. Both of the provided sample clients enable the logging. The log file used is named `cstadll_<program>.log`, where `<program>` is the name of the executable image that is using the DLL. The location of the log file is the folder where the executable image resides. If the log file grows to more than 5 Mb, it is copied to `cstadll_<program>.backup.log`, and a new log file is opened. If there is already a file with the backup file name, it is overwritten.

## Chapter 2

# Namespace Documentation

### 2.1 Package `Com.Objsys.Csta.Common`

#### Classes

- class [CSTAContext](#)
- class [CSTAResponseInfo](#)
- class [LicenseException](#)
- class [PBXSessionException](#)
- class [PBXSessionHelper](#)
- class [ROSEParseInfo](#)
- class [SocketState](#)

#### 2.1.1 Detailed Description

The namespace `Com.Objsys.Csta.Common` contains classes that are common to all phases.

## 2.2 Package Com.Objsys.Csta.Devices

### Classes

- class [Alcatel4400](#)
- class [AlcatelOXO](#)
- class [AlcatelOXOp1](#)
- class [PanasonicKXTDA](#)
- class [PanasonicKXTDE](#)
- class [PanasonicNCP](#)
- class [SiemensCap](#)
- class [SiemensHicom300](#)
- class [SiemensHipath3000p2](#)
- class [SiemensHipath3000p3](#)
- class [SiemensHipath4000](#)
- class [SiemensRealitis](#)
- class [TadiranCoral](#)
- class [UnifyOpenscapeX5](#)

### 2.2.1 Detailed Description

The namespace `Com.Objsys.Csta.Devices` contains classes that allow a caller to use specific PBX devices. The caller does not need to know what CSTA phase a device uses unless the device can accept messages formatted according to rules from more than one CSTA phase. In that case the class name ends with 'p(n)', where (n) is the number of the phase.



## 2.3 Package Com.Objsys.Csta.Phase1

### Classes

- class [GenericCSTAp1](#)
- class [IETF\\_CSTAp1](#)
- class [Phase1OpCodes](#)

### 2.3.1 Detailed Description

The namespace [Com.Objsys.Csta.Phase1](#) contains classes that are specific to phase 1. Most of these classes are generated by ASN1C from the CSTA and ACSE ASN.1 specifications. These generated classes are not documented here, but you can consult the ASN1C C# User Guide for information about how ASN.1 constructions are translated into C# classes.

The namespace also contains several classes that are not generated by ASN1C. These classes are the ones documented in this manual.

## 2.4 Package Com.Objsys.Csta.Phase2

### Classes

- class [GenericCSTAp2](#)
- class [IETF\\_CSTAp2](#)
- class [Phase2Opcodes](#)

### 2.4.1 Detailed Description

The namespace [Com.Objsys.Csta.Phase2](#) contains classes that are specific to phase 2. Most of these classes are generated by ASN1C from the CSTA and ACSE ASN.1 specifications. These generated classes are not documented here, but you can consult the ASN1C C# User Guide for information about how ASN.1 constructions are translated into C# classes.

The namespace also contains several classes that are not generated by ASN1C. These classes are the ones documented in this manual.

## 2.5 Package Com.Objsys.Csta.Phase3

### Classes

- class [GenericCSTAp3](#)
- class [IETF\\_CSTAp3](#)
- class [Phase3Opcodes](#)
- class [SingleStepTransferInfo](#)
- class [TransferCallInfo](#)

### 2.5.1 Detailed Description

The namespace [Com.Objsys.Csta.Phase3](#) contains classes that are specific to phase 3. Most of these classes are generated by ASN1C from the CSTA and ACSE ASN.1 specifications. These generated classes are not documented here, but you can consult the ASN1C C# User Guide for information about how ASN.1 constructions are translated into C# classes.

The namespace also contains several classes that are not generated by ASN1C. These classes are the ones documented in this manual.



# Chapter 3

## Class Documentation

### 3.1 Alcatel4400 Class Reference

Inherits [Com::Objsys::Csta::Phase2::IETF\\_CSTAp2](#).

#### Public Member Functions

- [Alcatel4400 \(\)](#)
- override [CSTARResponseInfo MakeACSEAssociation \(\)](#)

#### 3.1.1 Detailed Description

Implements CSTA phase 2 operations for the Alcatel 4400 PBX device.

#### 3.1.2 Constructor & Destructor Documentation

##### 3.1.2.1 Alcatel4400 ()

Default constructor.

#### 3.1.3 Member Function Documentation

##### 3.1.3.1 override CSTARResponseInfo MakeACSEAssociation () [virtual]

Establish an ACSE association with the PBX.

#### Returns

A CSTARResponseInfo object.

Reimplemented from [GenericCSTAp2](#).

## 3.2 AlcatelOXO Class Reference

Inherits [Com::Objsys::Csta::Phase2::IETF\\_CSTAp2](#).

### Public Member Functions

- [AlcatelOXO \(\)](#)
- override [CSTAResponseInfo MakeACSEAssociation \(\)](#)

### 3.2.1 Detailed Description

Implements CSTA phase 2 operations for the Alcatel OXO device.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 AlcatelOXO ()

Default constructor.

### 3.2.3 Member Function Documentation

#### 3.2.3.1 override [CSTAResponseInfo MakeACSEAssociation \(\)](#) [virtual]

Establish an ACSE association with the PBX.

#### Returns

A [CSTAResponseInfo](#) object.

Reimplemented from [GenericCSTAp2](#).

## 3.3 AlcatelOXOp1 Class Reference

Inherits [Com::Objsys::Csta::Phase1::IETF\\_CSTAp1](#).

### Public Member Functions

- [AlcatelOXOp1 \(\)](#)
- override [CSTARResponseInfo MakeACSEAssociation \(\)](#)

#### 3.3.1 Detailed Description

Implements CSTA phase 1 operations for the Alcatel OXO device.

#### 3.3.2 Constructor & Destructor Documentation

##### 3.3.2.1 AlcatelOXOp1 ()

Default constructor.

#### 3.3.3 Member Function Documentation

##### 3.3.3.1 override [CSTARResponseInfo MakeACSEAssociation \(\)](#) [virtual]

Establish an ACSE association with the PBX.

##### Returns

A [CSTARResponseInfo](#) object.

Reimplemented from [GenericCSTAp1](#).

## 3.4 CSTAContext Class Reference

### Properties

- `byte[] ResponseFromPBX` [get, set]
- `List<byte[]> ResponsesFromPBX` [get, set]

### 3.4.1 Detailed Description

The [CSTAContext](#) class contains information needed to manage the interaction between the thread and the PBX.

### 3.4.2 Property Documentation

#### 3.4.2.1 `byte [] ResponseFromPBX` [get, set]

See documentation for [CSTARResponseInfo.ResponseFromPBX](#).

#### 3.4.2.2 `List<byte[]> ResponsesFromPBX` [get, set]

See documentation for [CSTARResponseInfo.ResponsesFromPBX](#).



## 3.5 CSTAResponseInfo Class Reference

### Properties

- byte[] [ResponseFromPBX](#) [get, set]
- List< byte[] > [ResponsesFromPBX](#) [get, set]
- int [StatusCode](#) [get, set]
- string [StatusMessage](#) [get, set]

### 3.5.1 Detailed Description

Contains information about a PBX operation that was attempted.

### 3.5.2 Property Documentation

#### 3.5.2.1 byte [] ResponseFromPBX [get, set]

Contains the response from the PBX for messages that generate a single atomic response, or the immediate acknowledgement response for messages that generate multiple data responses (e.g., Get Switching Function [Devices](#)). If a message that normally generates multiple response segments encounters an error (e.g., the PBX rejects the message), then the single error message returned by the PBX will be in this property; the ResponsesFromPBX property will be null.

For CSTA operations this property is simply a reference to the ResponseFromPBX property of the thread's [CSTAContext](#) object. If the value of that property changes, then the value of this property changes.

#### 3.5.2.2 List<byte[]> ResponsesFromPBX [get, set]

Contains the responses from the PBX for messages that generate multiple response segments (e.g., Get Switching Function [Devices](#)). If such a message encounters an error (e.g., the PBX rejects the message), then the single error message returned by the PBX will be in the ResponseFromPBX property; this property will be null. In all cases the first response, which is the acknowledgement message from the PBX, will be in the ResponseFromPBX property.

This property is simply a reference to the ResponsesFromPBX property of the thread's [CSTAContext](#) object. If the value of that property changes, then the value of this property changes.

#### 3.5.2.3 int StatusCode [get, set]

A numeric status code. A value less than zero indicates that something went wrong during the attempted operation.

#### 3.5.2.4 string StatusMessage [get, set]

Text containing information about a PBX operation that has completed, either successfully or not.

## 3.6 GenericCSTAp1 Class Reference

Inherited by [SiemensHicom300](#), and [IETF\\_CSTAp1](#).

### Public Member Functions

- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer, string deviceToLift)
- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer)
- virtual [CSTARResponseInfo AnswerCall](#) (string deviceToLift)
- virtual [CSTARResponseInfo ClearConnection](#) (ConnectionID connToClear)
- virtual [CSTARResponseInfo ConsultationCall](#) (ConnectionID existingCall, string targetDevice)
- virtual [CSTARResponseInfo DivertCall](#) (string divertFrom, string divertTo)
- int [EncodeROSERequestHeader](#) ([CSTARResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer, [Phase1Opcodes.Opcodes](#) opcode)
- [GenericCSTAp1](#) ()
- virtual [CSTARResponseInfo MakeACSEAssociation](#) ()
- virtual [CSTARResponseInfo MakeCall](#) (string callingDevice, string calledDevice)
- virtual [CSTARResponseInfo MonitorStart](#) (string deviceToMonitor)
- virtual [CSTARResponseInfo MonitorStop](#) (string monitoredDevice)
- virtual [CSTARResponseInfo MonitorStop](#) (MonitorCrossRefID crossRefID)
- virtual [CSTARResponseInfo MonitorStop](#) (int crossRefInt)
- virtual [CSTARResponseInfo QueryDevice](#) (string deviceToQuery)
- virtual [CSTARResponseInfo ReleaseACSEAssociation](#) ()
- virtual [CSTARResponseInfo SnapshotDevice](#) (string deviceToSnapshot)
- virtual [CSTARResponseInfo TransferCall](#) (string heldDevice, string connectedDevice)
- virtual [CSTARResponseInfo TransferCall](#) (ConnectionID initiatedCall, ConnectionID originalCall)

### Properties

- [CSTAContext ThreadContext](#) [get]

### 3.6.1 Detailed Description

Implements CSTA phase 1 operations using BER.

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 [GenericCSTAp1](#) ()

Default constructor. Indicates that phase 1 is in effect.

### 3.6.3 Member Function Documentation

#### 3.6.3.1 [virtual CSTARResponseInfo AnswerCall](#) (ConnectionID *callToAnswer*, string *deviceToLift*) [virtual]

Answers a call.

## Parameters

*callToAnswer* The connection id of the call to answer.

*deviceToLift* The device (e.g., extension number) that is to answer the call.

## Returns

A CSTAResponseInfo object.

### 3.6.3.2 virtual CSTAResponseInfo AnswerCall (ConnectionID *callToAnswer*) [virtual]

Answers a call.

## Parameters

*callToAnswer* The ConnectionID of the call to answer.

## Returns

A CSTAResponseInfo object.

### 3.6.3.3 virtual CSTAResponseInfo AnswerCall (string *deviceToLift*) [virtual]

Answers a call.

## Parameters

*deviceToLift* The identification (e.g., phone number) of the device to answer.

## Returns

A CSTAResponseInfo object.

### 3.6.3.4 virtual CSTAResponseInfo ClearConnection (ConnectionID *connToClear*) [virtual]

Clears a connection.

## Parameters

*connToClear* The ConnectionID of the connection to clear.

## Returns

A CSTAResponseInfo object.

### 3.6.3.5 virtual CSTAResponseInfo ConsultationCall (ConnectionID *existingCall*, string *targetDevice*) [virtual]

Instruct the PBX to do a consultation call.

## Parameters

*existingCall* The connection id of the call for which the consultation call will be made.

*targetDevice* Identifier (e.g., phone number) of the device that is the target of the consultation call.

## Returns

A CSTAResponseInfo object.

### 3.6.3.6 virtual CSTAResponseInfo DivertCall (string *divertFrom*, string *divertTo*) [virtual]

Diverts a call from a source to a destination.

#### Parameters

*divertFrom* Identifier (e.g., phone number) of the call to be diverted.

*divertTo* Identifier (e.g., phone number) of the location to which the call is to be diverted.

#### Returns

A CSTAResponseInfo object.

### 3.6.3.7 int EncodeROSERequestHeader (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, Phase1Opcodes.Opcodes *opcode*)

This method prepends a ROSE header to an already encoded phase 1 CSTA message.

#### Parameters

*response* A CSTAResponseInfo object, used to communicate any exception information back to the caller.

*encodeBuffer* An Asn1BerEncodeBuffer instance containing the already encoded CSTA message.

*opcode* The opcode enumeration for the operation that the encoded CSTA message describes.

#### Returns

The length of the encoded message, including both the CSTA message and the ROSE header, or -1 if the encoding fails.

### 3.6.3.8 virtual CSTAResponseInfo MakeACSEAssociation () [virtual]

Establish an ACSE association with the PBX.

#### Returns

A CSTAResponseInfo object.

Reimplemented in [AlcatelOXOp1](#).

### 3.6.3.9 virtual CSTAResponseInfo MakeCall (string *callingDevice*, string *calledDevice*) [virtual]

Instruct the PBX to place a call.

#### Parameters

*callingDevice* Identifier (e.g., phone number) of the device making the call.

*calledDevice* Identifier (e.g., phone number) of the device being called.

#### Returns

A CSTAResponseInfo object.

### 3.6.3.10 virtual CSTAResponseInfo MonitorStart (string *deviceToMonitor*) [virtual]

Issues a MonitorStart request to the PBX.

#### Parameters

*deviceToMonitor* Identifier (e.g., telephone number) of the device to monitor.

#### Returns

A CSTAResponseInfo object.

### 3.6.3.11 virtual CSTAResponseInfo MonitorStop (string *monitoredDevice*) [virtual]

This method stops all monitors active against the indicated device, regardless of what thread started the monitor. The method will only stop monitors started through the [MonitorStart\(\)](#) method.

#### Parameters

*monitoredDevice* The monitored device (e.g., extension).

#### Returns

If no problems are encountered, the method returns a CSTAResponseInfo object containing the response from the PBX for the LAST Monitor Stop message.

If any problems are encountered, the method returns a CSTAResponseInfo object containing information about the error, including any response from the PBX for the problematic Monitor Stop message.

### 3.6.3.12 virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID *crossRefID*) [virtual]

Stop a previously started PBX monitor request.

#### Parameters

*crossRefID* The cross reference id of the monitor request as a MonitorCrossRefID object.

#### Returns

A CSTAResponseInfo object.

### 3.6.3.13 virtual CSTAResponseInfo MonitorStop (int *crossRefInt*) [virtual]

Stop a previously started PBX monitor request.

#### Parameters

*crossRefInt* The cross reference id of the monitor request as an integer.

#### Returns

A CSTAResponseInfo object.

#### **3.6.3.14 virtual CSTAResponseInfo QueryDevice (string *deviceToQuery*) [virtual]**

Queries a device.

##### **Parameters**

*deviceToQuery* The identification (e.g., phone number) of the device to query.

##### **Returns**

A CSTAResponseInfo object.

#### **3.6.3.15 virtual CSTAResponseInfo ReleaseACSEAssociation () [virtual]**

Releases an ACSE association with a PBX device.

##### **Returns**

A CSTAResponseInfo object. For this message the connection with the PBX is closed, so null is returned.

#### **3.6.3.16 virtual CSTAResponseInfo SnapshotDevice (string *deviceToSnapshot*) [virtual]**

Instruct the PBX to take a snapshot of calls active at a device.

##### **Parameters**

*deviceToSnapshot* Identifier (e.g., phone number) of the device for which the snapshot is desired.

##### **Returns**

A CSTAResponseInfo object.

#### **3.6.3.17 virtual CSTAResponseInfo TransferCall (string *heldDevice*, string *connectedDevice*) [virtual]**

Transfers a call from one device to another.

##### **Parameters**

*heldDevice* Identifier (e.g., phone number) of the device from which the call is transferred.

*connectedDevice* Identifier (e.g., phone number) of the device to which the call is transferred.

##### **Returns**

A CSTAResponseInfo object.

#### **3.6.3.18 virtual CSTAResponseInfo TransferCall (ConnectionID *initiatedCall*, ConnectionID *originalCall*) [virtual]**

Transfers a call. A consultation call must be done before calling this method.

##### **Parameters**

*initiatedCall* ConnectionID of the new call initiated by the consultation call. The *initiatedCall* member of the ConsultationCallResult class, for example, contains this ConnectionID.

*originalCall* ConnectionID of the original call. The somewhat confusingly named *callingDevice* member of the *MakeCallResult* class contains this ConnectionID, as does the *establishedConnection* member of the *EstablishedEvent* class.

**Returns**

A *CSTARResponseInfo* object.

**3.6.4 Property Documentation**

**3.6.4.1 CSTAContext ThreadContext [get]**

The *CSTAContext* structure for this thread.

## 3.7 GenericCSTAp2 Class Reference

Inherited by [SiemensHipath3000p2](#), and [IETF\\_CSTAp2](#).

### Public Member Functions

- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer, string deviceToLift)
- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer)
- virtual [CSTARResponseInfo AnswerCall](#) (string deviceToLift)
- virtual [CSTARResponseInfo ClearConnection](#) (ConnectionID connToClear)
- virtual [CSTARResponseInfo ConsultationCall](#) (ConnectionID existingCall, string targetDevice)
- virtual [CSTARResponseInfo DivertCall](#) (string divertFrom, string divertTo)
- int [EncodeROSERequestHeader](#) (CSTARResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, Phase2Opcodes.Opcodes opcode)
- [GenericCSTAp2](#) ()
- virtual [CSTARResponseInfo MakeACSEAssociation](#) ()
- virtual [CSTARResponseInfo MakeCall](#) (string callingDevice, string calledDevice)
- virtual [CSTARResponseInfo MonitorStart](#) (string deviceToMonitor)
- virtual [CSTARResponseInfo MonitorStop](#) (string monitoredDevice)
- virtual [CSTARResponseInfo MonitorStop](#) (MonitorCrossRefID crossRefID)
- virtual [CSTARResponseInfo MonitorStop](#) (int crossRefInt)
- virtual [CSTARResponseInfo QueryDevice](#) (string deviceToQuery)
- virtual [CSTARResponseInfo ReleaseACSEAssociation](#) ()
- virtual [CSTARResponseInfo SnapshotDevice](#) (string deviceToSnapshot)
- virtual [CSTARResponseInfo TransferCall](#) (string heldDevice, string connectedDevice)
- virtual [CSTARResponseInfo TransferCall](#) (ConnectionID initiatedCall, ConnectionID originalCall)

### Properties

- [CSTAContext ThreadContext](#) [get]

#### 3.7.1 Detailed Description

Implements CSTA phase 2 operations using BER.

#### 3.7.2 Constructor & Destructor Documentation

##### 3.7.2.1 [GenericCSTAp2](#) ()

Default constructor. Indicates that phase 2 is in effect.

#### 3.7.3 Member Function Documentation

##### 3.7.3.1 [virtual CSTARResponseInfo AnswerCall](#) (ConnectionID *callToAnswer*, string *deviceToLift*) [virtual]

Answers a call.



## Parameters

*callToAnswer* The connection id of the call to answer.

*deviceToLift* The device (e.g., extension number) that is to answer the call.

## Returns

A CSTAResponseInfo object.

### 3.7.3.2 virtual CSTAResponseInfo AnswerCall (ConnectionID *callToAnswer*) [virtual]

Answers a call.

## Parameters

*callToAnswer* The ConnectionID of the call to answer.

## Returns

A CSTAResponseInfo object.

### 3.7.3.3 virtual CSTAResponseInfo AnswerCall (string *deviceToLift*) [virtual]

Answers a call.

## Parameters

*deviceToLift* The identification (e.g., phone number) of the device to answer.

## Returns

A CSTAResponseInfo object.

### 3.7.3.4 virtual CSTAResponseInfo ClearConnection (ConnectionID *connToClear*) [virtual]

Clears a connection.

## Parameters

*connToClear* The ConnectionID of the connection to clear.

## Returns

A CSTAResponseInfo object.

### 3.7.3.5 virtual CSTAResponseInfo ConsultationCall (ConnectionID *existingCall*, string *targetDevice*) [virtual]

Instruct the PBX to do a consultation call.

## Parameters

*existingCall* The connection id of the call for which the consultation call will be made.

*targetDevice* Identifier (e.g., phone number) of the device that is the target of the consultation call.

## Returns

A CSTAResponseInfo object.

### 3.7.3.6 virtual CSTAResponseInfo DivertCall (string *divertFrom*, string *divertTo*) [virtual]

Diverts a call from a source to a destination.

#### Parameters

*divertFrom* Identifier (e.g., phone number) of the call to be diverted.

*divertTo* Identifier (e.g., phone number) of the location to which the call is to be diverted.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.7 int EncodeROSERequestHeader (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, Phase2Opcodes.Opcodes *opcode*)

This method prepends a ROSE header to an already encoded phase 2 CSTA message.

#### Parameters

*response* A CSTAResponseInfo object, used to communicate any exception information back to the caller.

*encodeBuffer* An Asn1BerEncodeBuffer instance containing the already encoded CSTA message.

*opcode* The opcode enumeration for the operation that the encoded CSTA message describes.

#### Returns

The length of the encoded message, including both the CSTA message and the ROSE header, or -1 if the encoding fails.

### 3.7.3.8 virtual CSTAResponseInfo MakeACSEAssociation () [virtual]

Establish an ACSE association with the PBX.

#### Returns

A CSTAResponseInfo object.

Reimplemented in [Alcatel4400](#), and [AlcatelOXO](#).

### 3.7.3.9 virtual CSTAResponseInfo MakeCall (string *callingDevice*, string *calledDevice*) [virtual]

Instruct the PBX to place a call.

#### Parameters

*callingDevice* Identifier (e.g., phone number) of the device making the call.

*calledDevice* Identifier (e.g., phone number) of the device being called.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.10 virtual CSTAResponseInfo MonitorStart (string *deviceToMonitor*) [virtual]

Issues a MonitorStart request to the PBX.

#### Parameters

*deviceToMonitor* Identifier (e.g., telephone number) of the device to monitor.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.11 virtual CSTAResponseInfo MonitorStop (string *monitoredDevice*) [virtual]

This method stops all monitors active against the indicated device, regardless of what thread started the monitor. The method will only stop monitors started through the [MonitorStart\(\)](#) method.

#### Parameters

*monitoredDevice* The monitored device (e.g., extension).

#### Returns

If no problems are encountered, the method returns a CSTAResponseInfo object containing the response from the PBX for the LAST Monitor Stop message.

If any problems are encountered, the method returns a CSTAResponseInfo object containing information about the error, including any response from the PBX for the problematic Monitor Stop message.

### 3.7.3.12 virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID *crossRefID*) [virtual]

Stop a previously started PBX monitor request.

#### Parameters

*crossRefID* The cross reference id of the monitor request as a MonitorCrossRefID object.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.13 virtual CSTAResponseInfo MonitorStop (int *crossRefInt*) [virtual]

Stop a previously started PBX monitor request.

#### Parameters

*crossRefInt* The cross reference id of the monitor request as an integer.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.14 virtual CSTAResponseInfo QueryDevice (string *deviceToQuery*) [virtual]

Queries a device.

#### Parameters

*deviceToQuery* The identification (e.g., phone number) of the device to query.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.15 virtual CSTAResponseInfo ReleaseACSEAssociation () [virtual]

Releases an ACSE association with a PBX device.

#### Returns

A CSTAResponseInfo object. For this message the connection with the PBX is closed, so null is returned.

### 3.7.3.16 virtual CSTAResponseInfo SnapshotDevice (string *deviceToSnapshot*) [virtual]

Instruct the PBX to take a snapshot of calls active at a device.

#### Parameters

*deviceToSnapshot* Identifier (e.g., phone number) of the device for which the snapshot is desired.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.17 virtual CSTAResponseInfo TransferCall (string *heldDevice*, string *connectedDevice*) [virtual]

Transfers a call from one device to another.

#### Parameters

*heldDevice* Identifier (e.g., phone number) of the device from which the call is transferred.

*connectedDevice* Identifier (e.g., phone number) of the device to which the call is transferred.

#### Returns

A CSTAResponseInfo object.

### 3.7.3.18 virtual CSTAResponseInfo TransferCall (ConnectionID *initiatedCall*, ConnectionID *originalCall*) [virtual]

Transfers a call. A consultation call must be done before calling this method.

#### Parameters

*initiatedCall* ConnectionID of the new call initiated by the consultation call. The *initiatedCall* member of the ConsultationCallResult class, for example, contains this ConnectionID.

*originalCall* ConnectionID of the original call. The somewhat confusingly named *callingDevice* member of the *MakeCallResult* class contains this ConnectionID, as does the *establishedConnection* member of the *EstablishedEvent* class.

#### **Returns**

A *CSTARResponseInfo* object.

### **3.7.4 Property Documentation**

#### **3.7.4.1 CSTAContext ThreadContext [get]**

The *CSTAContext* structure for this thread.

## 3.8 GenericCSTAp3 Class Reference

Inherited by [SiemensHipath3000p3](#), and [IETF\\_CSTAp3](#).

### Public Member Functions

- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer, string deviceToLift)
- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer)
- virtual [CSTARResponseInfo AnswerCall](#) (string deviceToLift)
- virtual [CSTARResponseInfo ClearConnection](#) (ConnectionID connToClear)
- virtual [CSTARResponseInfo ClearMessageWaiting](#) (string targetDevice)
- virtual [CSTARResponseInfo ConsultationCall](#) (ConnectionID existingCall, string targetDevice)
- virtual int [EncodeROSERequestHeader](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, [Phase3Opcodes.Opcodes](#) opcode)
- [GenericCSTAp3](#) ()
- virtual [CSTARResponseInfo GetSFDevices](#) (ReqDeviceCategory deviceCategory)
- virtual [CSTARResponseInfo GetSFDevices](#) ()
- virtual [CSTARResponseInfo MakeACSEAssociation](#) ()
- virtual [CSTARResponseInfo MakeCall](#) (string callingDevice, string calledDevice)
- virtual [CSTARResponseInfo MonitorStart](#) (string deviceToMonitor)
- virtual [CSTARResponseInfo MonitorStop](#) (string monitoredDevice)
- virtual [CSTARResponseInfo MonitorStop](#) (MonitorCrossRefID crossRefID)
- virtual [CSTARResponseInfo MonitorStop](#) (int crossRefInt)
- virtual [CSTARResponseInfo ReleaseACSEAssociation](#) ()
- virtual [CSTARResponseInfo RingDevice](#) (string targetDevice, string targetRinger, long ringPattern)
- virtual [CSTARResponseInfo SendData](#) (IOCrossRefID ioCrossRef, string text)
- virtual [CSTARResponseInfo SetDisplay](#) (string targetDevice, string text)
- virtual [CSTARResponseInfo SetMessageWaiting](#) (string targetDevice)
- virtual [CSTARResponseInfo SingleStepTransfer](#) ([SingleStepTransferInfo](#) sstInfo)
- virtual [CSTARResponseInfo SingleStepTransfer](#) (ConnectionID callToTransfer, string transferToDevice)
- virtual [CSTARResponseInfo SnapshotDevice](#) (string deviceToSnapshot)
- virtual [CSTARResponseInfo StartDataPath](#) (string targetDevice)
- virtual [CSTARResponseInfo StopDataPath](#) (IOCrossRefID ioCrossRef)
- virtual [CSTARResponseInfo StopRing](#) (string targetDevice, string targetRinger, long ringPattern)
- virtual [CSTARResponseInfo TransferCall](#) ([TransferCallInfo](#) tcInfo)
- virtual [CSTARResponseInfo TransferCall](#) (ConnectionID initiatedCall, ConnectionID originalCall)

### Properties

- [CSTAContext ThreadContext](#) [get]

### 3.8.1 Detailed Description

Implements CSTA phase 3 operations using BER.

### 3.8.2 Constructor & Destructor Documentation

#### 3.8.2.1 [GenericCSTAp3](#) ()

Default constructor. Indicates that phase 3 is in effect.

### 3.8.3 Member Function Documentation

#### 3.8.3.1 virtual CSTAResponseInfo AnswerCall (ConnectionID *callToAnswer*, string *deviceToLift*) [virtual]

Answers a call.

##### Parameters

*callToAnswer* ConnectionID of an existing call (such as initiated through [MakeCall\(\)](#)).

*deviceToLift* The device (e.g., "800") that is to answer the call.

##### Returns

A CSTAResponseInfo object.

#### 3.8.3.2 virtual CSTAResponseInfo AnswerCall (ConnectionID *callToAnswer*) [virtual]

Answers a call.

##### Parameters

*callToAnswer* The ConnectionID of the call to answer.

##### Returns

A CSTAResponseInfo object.

#### 3.8.3.3 virtual CSTAResponseInfo AnswerCall (string *deviceToLift*) [virtual]

Answers a call.

##### Parameters

*deviceToLift* The identification (e.g., phone number) of the device to answer.

##### Returns

A CSTAResponseInfo object.

#### 3.8.3.4 virtual CSTAResponseInfo ClearConnection (ConnectionID *connToClear*) [virtual]

Clears a connection.

##### Parameters

*connToClear* The ConnectionID of the connection to clear.

##### Returns

A CSTAResponseInfo object.

### 3.8.3.5 virtual CSTAResponseInfo ClearMessageWaiting (string *targetDevice*) [virtual]

Turns off the message waiting indicator on a device's display.

#### Parameters

*targetDevice* The device for which the indicator is to be turned off.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.6 virtual CSTAResponseInfo ConsultationCall (ConnectionID *existingCall*, string *targetDevice*) [virtual]

Instruct the PBX to do a consultation call.

#### Parameters

*existingCall* The connection id of the call for which the consultation call will be made.

*targetDevice* Identifier (e.g., phone number) of the device that is the target of the consultation call.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.7 virtual int EncodeROSERequestHeader (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, Phase3OpCodes.OpCodes *opcode*) [virtual]

This method prepends a ROSE header to an already encoded phase 3 CSTA message.

#### Parameters

*response* A CSTAResponseInfo object, used to communicate any exception information back to the caller.

*encodeBuffer* An Asn1BerEncodeBuffer instance containing the already encoded CSTA message.

*opcode* The opcode enumeration for the operation that the encoded CSTA message describes.

#### Returns

The length of the encoded message, including both the CSTA message and the ROSE header, or -1 if the encoding fails.

### 3.8.3.8 virtual CSTAResponseInfo GetSFDevices (ReqDeviceCategory *deviceCategory*) [virtual]

Sends a Get Switching Function [Devices](#) request to the PBX.

#### Parameters

*deviceCategory* The category of device for which the list is desired.

#### Returns

A CSTAResponseInfo object.



### 3.8.3.9 virtual CSTAResponseInfo GetSFDevices () [virtual]

Sends a Get Switching Function [Devices](#) request to the PBX.

#### Returns

A CSTAResponseInfo object.

Reimplemented in [PanasonicKXTDE](#).

### 3.8.3.10 virtual CSTAResponseInfo MakeACSEAssociation () [virtual]

Establish an ACSE association with the PBX.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.11 virtual CSTAResponseInfo MakeCall (string *callingDevice*, string *calledDevice*) [virtual]

Instruct the PBX to place a call.

#### Parameters

*callingDevice* Identifier (e.g., phone number) of the device making the call.

*calledDevice* Identifier (e.g., phone number) of the device being called.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.12 virtual CSTAResponseInfo MonitorStart (string *deviceToMonitor*) [virtual]

Issues a MonitorStart request to the PBX.

#### Parameters

*deviceToMonitor* Identifier (e.g., telephone number) of the device to monitor.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.13 virtual CSTAResponseInfo MonitorStop (string *monitoredDevice*) [virtual]

This method stops all monitors active against the indicated device, regardless of what thread started the monitor. The method will only stop monitors started through the [MonitorStart\(\)](#) method.

#### Parameters

*monitoredDevice* The monitored device (e.g., extension).

## Returns

If no problems are encountered, the method returns a `CSTARResponseInfo` object containing the response from the PBX for the LAST Monitor Stop message.

If any problems are encountered, the method returns a `CSTARResponseInfo` object containing information about the error, including any response from the PBX for the problematic Monitor Stop message.

### 3.8.3.14 virtual `CSTARResponseInfo MonitorStop (MonitorCrossRefID crossRefID)` [virtual]

Stop a previously started PBX monitor request.

#### Parameters

*crossRefID* The cross reference id of the monitor request as a `MonitorCrossRefID` object.

#### Returns

A `CSTARResponseInfo` object.

### 3.8.3.15 virtual `CSTARResponseInfo MonitorStop (int crossRefInt)` [virtual]

Stop a previously started PBX monitor request.

#### Parameters

*crossRefInt* The cross reference id of the monitor request as an integer.

#### Returns

A `CSTARResponseInfo` object.

### 3.8.3.16 virtual `CSTARResponseInfo ReleaseACSEAssociation ()` [virtual]

Releases an ACSE association with a PBX device.

#### Returns

A `CSTARResponseInfo` object. For this message the connection with the PBX is closed, so null is returned.

### 3.8.3.17 virtual `CSTARResponseInfo RingDevice (string targetDevice, string targetRinger, long ringPattern)` [virtual]

Causes a telephony device to ring.

#### Parameters

*targetDevice* The device to ring.

*targetRinger* The id of the ringer to use for the ring. This argument can be specified as a character string (e.g, "abc"), a hex string (e.g, "010A05'H"), or a binary string (e.g, "000000010000101000000101'B").

*ringPattern* The indicator of the ring pattern to use.

#### Returns

A `CSTARResponseInfo` object.

### 3.8.3.18 virtual CSTAResponseInfo SendData (IOCrossRefID *ioCrossRef*, string *text*) [virtual]

Sends a text message to a telephony device.

#### Parameters

*ioCrossRef* An IOCrossRefID object, most likely obtained by a previous call to StartDataPath.

*text* The text to send to the telephony device.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.19 virtual CSTAResponseInfo SetDisplay (string *targetDevice*, string *text*) [virtual]

Sends text to a telephony device's display

#### Parameters

*targetDevice* The device to which the text is to be sent.

*text* The text to be sent.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.20 virtual CSTAResponseInfo SetMessageWaiting (string *targetDevice*) [virtual]

Turns on the message waiting indicator on a device's display.

#### Parameters

*targetDevice* The device for which the indicator is to be turned on.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.21 virtual CSTAResponseInfo SingleStepTransfer (SingleStepTransferInfo *sstInfo*) [virtual]

Perform a single step transfer.

#### Parameters

*sstInfo* A [SingleStepTransferInfo](#) object.

#### Returns

A CSTAResponseInfo object.

**3.8.3.22 virtual CSTAResponseInfo SingleStepTransfer (ConnectionID *callToTransfer*, string *transferToDevice*) [virtual]**

Perform a single step transfer.

**Parameters**

*callToTransfer* The ConnectionID of the call to transfer.

*transferToDevice* The device to which the call is to be transferred.

**Returns**

A CSTAResponseInfo object.

**3.8.3.23 virtual CSTAResponseInfo SnapshotDevice (string *deviceToSnapshot*) [virtual]**

Instruct the PBX to take a snapshot of calls active at a device.

**Parameters**

*deviceToSnapshot* Identifier (e.g., phone number) of the device for which the snapshot is desired.

**Returns**

A CSTAResponseInfo object.

**3.8.3.24 virtual CSTAResponseInfo StartDataPath (string *targetDevice*) [virtual]**

Opens up a data path to a specified device.

**Parameters**

*targetDevice* Specifies the device to which a data path is to be opened.

**Returns**

A CSTAResponseInfo object.

**3.8.3.25 virtual CSTAResponseInfo StopDataPath (IOCrossRefID *ioCrossRef*) [virtual]**

Stops a previously established data path

**Parameters**

*ioCrossRef* An IOCrossRef object, most likely obtained from a previous call to StartDataPath.

**Returns**

A CSTAResponseInfo object.

### 3.8.3.26 virtual CSTAResponseInfo StopRing (string *targetDevice*, string *targetRinger*, long *ringPattern*) [virtual]

Stops a ringer on a telephony device.

#### Parameters

*targetDevice* The device for which the ringer is to stop.

*targetRinger* The id of the ringer to stop. This argument can be specified as a character string (e.g, "abc"), a hex string (e.g, "010A05'H"), or a binary string (e.g, "000000010000101000000101'B").

*ringPattern* The indicator of the ring pattern to stop.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.27 virtual CSTAResponseInfo TransferCall (TransferCallInfo *tcInfo*) [virtual]

Transfers a call from one device to another.

#### Parameters

*tcInfo* A [TransferCallInfo](#) object.

#### Returns

A CSTAResponseInfo object.

### 3.8.3.28 virtual CSTAResponseInfo TransferCall (ConnectionID *initiatedCall*, ConnectionID *originalCall*) [virtual]

Transfers a call. A consultation call must be done before calling this method.

#### Parameters

*initiatedCall* ConnectionID of the new call initiated by the consultation call. The *initiatedCall* member of the [ConsultationCallResult](#) class, for example, contains this ConnectionID.

*originalCall* ConnectionID of the original call. The somewhat confusingly named *callingDevice* member of the [MakeCallResult](#) class contains this ConnectionID, as does the *establishedConnection* member of the [EstablishedEvent](#) class.

#### Returns

A CSTAResponseInfo object.

## 3.8.4 Property Documentation

### 3.8.4.1 CSTAContext ThreadContext [get]

The CSTAContext structure for this thread.

## 3.9 IETF\_CSTAp1 Class Reference

Inherits [Com::Objsys::Csta::Phase1::GenericCSTAp1](#).

Inherited by [AlcatelOXOp1](#), [SiemensRealitis](#), and [TadiranCoral](#).

### Public Member Functions

- [IETF\\_CSTAp1 \(\)](#)

#### 3.9.1 Detailed Description

Implements CSTA phase 1 operations using IETF encoding, which puts a two-byte length in front of the BER message.

#### 3.9.2 Constructor & Destructor Documentation

##### 3.9.2.1 IETF\_CSTAp1 ()

Default constructor. Indicates to the PBXSession that IETF is in effect.

## 3.10 IETF\_CSTAp2 Class Reference

Inherits [Com::Objsys::Csta::Phase2::GenericCSTAp2](#).

Inherited by [Alcatel4400](#), and [AlcatelOXO](#).

### Public Member Functions

- [IETF\\_CSTAp2 \(\)](#)

#### 3.10.1 Detailed Description

Implements CSTA phase 2 operations using IETF encoding, which puts a two-byte length in front of the BER message.

#### 3.10.2 Constructor & Destructor Documentation

##### 3.10.2.1 IETF\_CSTAp2 ()

Default constructor. Indicates to the PBXSession that IETF is in effect.

## 3.11 IETF\_CSTAp3 Class Reference

Inherits [Com::Objsys::Csta::Phase3::GenericCSTAp3](#).

Inherited by [PanasonicKXTDE](#), and [SiemensCap](#).

### Public Member Functions

- [IETF\\_CSTAp3](#) ()

#### 3.11.1 Detailed Description

Implements CSTA phase 3 operations using IETF encoding, which puts a two-byte length in front of the BER message.

#### 3.11.2 Constructor & Destructor Documentation

##### 3.11.2.1 IETF\_CSTAp3 ()

Default constructor. Indicates to the PBXSession that IETF is in effect.



## **3.12 LicenseException Class Reference**

### **3.12.1 Detailed Description**

Defines an exception that occurs while trying to find license information.

## **3.13 PanasonicKXTDA Class Reference**

Inherits [Com::Objsys::Csta::Devices::PanasonicKXTDE](#).

### **3.13.1 Detailed Description**

Implements CSTA phase 3 operations for the Panasonic KX-TDA PBX device.

## 3.14 PanasonicKXTDE Class Reference

Inherits [Com::Objsys::Csta::Phase3::IETF\\_CSTAp3](#).

Inherited by [PanasonicKXTDA](#), and [PanasonicNCP](#).

### Public Member Functions

- [CSTARResponseInfo AcquireControlRight](#) (string targetDevice)
- [CSTARResponseInfo ClearMessageWaiting](#) (string originatingDevice, string targetDevice)
- [CSTARResponseInfo GetGroupMembers](#) (string groupDevice)
- override [CSTARResponseInfo GetSFDevices](#) ()
- [CSTARResponseInfo PDFStart](#) (string targetDevice)
- [CSTARResponseInfo PDFStop](#) (string targetDevice)
- [CSTARResponseInfo ReleaseControlRight](#) (string targetDevice)
- [CSTARResponseInfo ResetDisplay](#) (string targetDevice)
- [CSTARResponseInfo SendKmeMessage](#) (Asn1BerEncodeBuffer encodeBuffer)
- [CSTARResponseInfo SetMessageWaiting](#) (string originatingDevice, string targetDevice)

### 3.14.1 Detailed Description

Implements CSTA phase 3 operations for the Panasonic KX-TDE PBX device.

### 3.14.2 Member Function Documentation

#### 3.14.2.1 [CSTARResponseInfo AcquireControlRight](#) (string *targetDevice*)

Acquires the right to control a telephony device. This method will cause a "PDF (Physical Device Feature) Start" Escape message to be sent to the PBX.

#### Parameters

*targetDevice* The target telephony device.

#### Returns

A [CSTARResponseInfo](#) object.

#### 3.14.2.2 [CSTARResponseInfo ClearMessageWaiting](#) (string *originatingDevice*, string *targetDevice*)

Turns off a device's message waiting indicator.

#### Parameters

*originatingDevice* The device that originated the call back request.

*targetDevice* The device for which the message waiting indicator is to be turned off.

#### Returns

A [CSTARResponseInfo](#) object.

### 3.14.2.3 **CSTARResponseInfo GetGroupMembers (string *groupDevice*)**

Gets the members associated with a group device (for example, the extensions associated with an incoming call distribution group device).

#### **Parameters**

*groupDevice* The group device designation (e.g, "601").

#### **Returns**

A CSTARResponseInfo object.

### 3.14.2.4 **override CSTARResponseInfo GetSFDevices () [virtual]**

Returns a list of station (i.e., telephone) devices known to the PBX by sending a Get Switching Function [Devices](#) message that specifies just station devices.

#### **Returns**

A CSTARResponseInfo object.

Reimplemented from [GenericCSTAp3](#).

### 3.14.2.5 **CSTARResponseInfo PDFStart (string *targetDevice*)**

Sends a "PDF (Physical Device Feature) Start" Escape message to the PBX. This method is a convenience method that does the same thing as the [AcquireControlRight\(\)](#) method.

#### **Parameters**

*targetDevice* The target telephony device.

#### **Returns**

A CSTARResponseInfo object.

### 3.14.2.6 **CSTARResponseInfo PDFStop (string *targetDevice*)**

Sends a "PDF (Physical Device Feature) Stop" Escape message to the PBX. This method is a convenience method that does the same thing as the [ReleaseControlRight\(\)](#) method.

#### **Parameters**

*targetDevice* The target telephony device.

#### **Returns**

A CSTARResponseInfo object.

### 3.14.2.7 **CSTARResponseInfo ReleaseControlRight (string *targetDevice*)**

Releases the right to control a telephony device. This method will cause a "PDF (Physical Device Feature) Stop" Escape message to be sent to the PBX.

#### **Parameters**

*targetDevice* The target telephony device.

#### **Returns**

A CSTARResponseInfo object.

### 3.14.2.8 **CSTARResponseInfo ResetDisplay (string *targetDevice*)**

Resets a telephony device's display

#### **Parameters**

*targetDevice* The device which is to be reset.

#### **Returns**

A CSTARResponseInfo object.

### 3.14.2.9 **CSTARResponseInfo SendKmeMessage (Asn1BerEncodeBuffer *encodeBuffer*)**

Sends a Panasonic-specific KME message to the PBX.

#### **Parameters**

*encodeBuffer* A BER encode buffer instance containing a completely encoded KME message. The message must include the KMESpecificPrivateData segment, the EscapeArgument, and the ROSE header.

#### **Returns**

### 3.14.2.10 **CSTARResponseInfo SetMessageWaiting (string *originatingDevice*, string *targetDevice*)**

Turns on a device's message waiting indicator.

#### **Parameters**

*originatingDevice* The device that originated the call back request.

*targetDevice* The device for which the message waiting indicator is to be turned on.

#### **Returns**

A CSTARResponseInfo object.

## **3.15 PanasonicNCP Class Reference**

Inherits [Com::Objsys::Csta::Devices::PanasonicKXTDE](#).

### **3.15.1 Detailed Description**

Implements CSTA phase 3 operations for the Panasonic NCP PBX device.

## **3.16 PBXSessionException Class Reference**

### **3.16.1 Detailed Description**

Defines an exception that occurs while communicating with a PBX.

## 3.17 PBXSessionHelper Class Reference

### Public Member Functions

- delegate void [AsyncCallback](#) (byte[] asyncData)
- delegate void [ConnectionCallback](#) ()

### Static Public Member Functions

- static void [Close](#) ([CSTAContext](#) threadContext)
- static void [Init](#) (string pbxSystem, int port)
- static void [Open](#) ([CSTAContext](#) threadContext)
- static [SocketState](#) [SendACSEMessage](#) (byte[] message, int messageLength, Constants.ACSEMessageTypes messageType, [CSTAContext](#) threadContext)
- static void [SendMessage](#) (string messageType, byte[] message, int messageLength, [CSTAContext](#) threadContext)
- static void [SendMessage](#) (byte[] message, int messageLength, [CSTAContext](#) threadContext)
- static void [WaitForROSEResponse](#) ([CSTAContext](#) threadContext)

### Properties

- static AsyncCallback [ClientCallback](#) [get, set]
- static ConnectionCallback [ConnectionLostCallback](#) [get, set]
- static bool [DebugMode](#) [get, set]
- static bool [LoggingEnabled](#) [get, set]
- static string [LoggingFolder](#) [get, set]

#### 3.17.1 Detailed Description

Manages the communication with the PBX. This class is a static helper class that allows a client of the DLL to communicate with a single PBX.

#### 3.17.2 Member Function Documentation

##### 3.17.2.1 delegate void AsyncCallback (byte[] asyncData)

Declaration of a callback function to be invoked when an asynchronous message is received, such as from a monitor session.

##### Parameters

*asyncData* The data received asynchronously from the PBX.

##### 3.17.2.2 static void Close (CSTAContext threadContext) [static]

Terminates the session to the PBX. This method can be used to terminate sessions with PBX devices that don't accept ACSE release association requests.

##### Parameters

*threadContext* The context object for the calling thread.



### 3.17.2.3 **delegate void ConnectionCallback ()**

Declaration of a callback function to be invoked if the connection to the PBX is lost.

### 3.17.2.4 **static void Init (string *pbxSystem*, int *port*) [static]**

Initializes the PBX Session.

#### **Parameters**

*pbxSystem* The name or IP address of the PBX system.

*port* The port on the PBX system to which the client is connecting.

### 3.17.2.5 **static void Open (CSTAContext *threadContext*) [static]**

This method can be used to establish communication with a PBX device before any messages are actually sent to the device.

#### **Parameters**

*threadContext* The thread context object.

### 3.17.2.6 **static SocketState SendACSEMessage (byte[] *message*, int *messageLength*, Constants.ACSEMessageTypes *messageType*, CSTAContext *threadContext*) [static]**

This method sends an ACSE message (either Make Association or Release Association) to the PBX and receives the response. This operation is done synchronously. If the Make Association needs to be done (usually it does), it must be done before any threads for sending and receiving CSTA messages are started.

This method is only intended to be used by client code that encodes its own ACSEMakeAssociation or ACSEReleaseAssociation message. Most clients can probably use the MakeACSEAssociation() and ReleaseACSEAssociation() methods that are in each phase's helper classes.

#### **Parameters**

*message* An encoded ACSE Make Association or Release Association message.

*messageLength* The length of the encoded message.

*messageType* A constant telling whether the message is an ACSE Make Association or an ACSE Release Association.

*threadContext* The thread context object.

#### **Returns**

A populated [SocketState](#) instance.

### 3.17.2.7 **static void SendMessage (string *messageType*, byte[] *message*, int *messageLength*, CSTAContext *threadContext*) [static]**

This method sends a message to the PBX using TCP/IP.

## Parameters

*messageType* A string token to help identify the message in the CSTADLL log file.

*message* Byte array containing the encoded message to send.

*messageLength* The length of the encoded message.

*threadContext* The thread context object.

### 3.17.2.8 `static void SendMessage (byte[] message, int messageLength, CSTAContext threadContext)` `[static]`

This method sends a message to the PBX using TCP/IP.

## Parameters

*message* Byte array containing the encoded message to send.

*messageLength* The length of the encoded message.

*threadContext* The thread context object.

### 3.17.2.9 `static void WaitForROSEResponse (CSTAContext threadContext)` `[static]`

This method waits for a response to a CSTA message sent with a ROSE header.

## Parameters

*threadContext* The [CSTAContext](#) object associated with the calling thread.

## 3.17.3 Property Documentation

### 3.17.3.1 `AsyncCallback ClientCallback` `[static, get, set]`

Holds a reference to an asynchronous callback function. This function will be invoked if data is received asynchronously from the PBX, such as from a monitor operation.

### 3.17.3.2 `ConnectionCallback ConnectionLostCallback` `[static, get, set]`

Holds a reference to an asynchronous callback function. This function will be invoked if the connection to the PBX is lost.

### 3.17.3.3 `bool DebugMode` `[static, get, set]`

Enables behavior that facilitates debugging of the CSTADLL software. This property is most likely useful only to Objective Systems staff.

### 3.17.3.4 `bool LoggingEnabled` `[static, get, set]`

Determines whether logging of traffic between the client and the PBX will be done.

### 3.17.3.5 string LoggingFolder [static, get, set]

Specifies a folder to receive the log file. If not specified, the log file will go into whatever folder the calling .exe resides in.

## 3.18 Phase1Opcodes Class Reference

### Public Types

- enum [Opcodes](#)

#### 3.18.1 Detailed Description

This class contains a public enum that contains symbolic names for the opcodes that define CSTA phase 1 operations.

#### 3.18.2 Member Enumeration Documentation

##### 3.18.2.1 enum Opcodes

Contains symbolic names for the opcodes that define CSTA phase 1 operations.

## 3.19 Phase2Opcodes Class Reference

### Public Types

- enum [Opcodes](#)

#### 3.19.1 Detailed Description

This class contains a public enum that contains symbolic names for the opcodes that define CSTA phase 2 operations.

#### 3.19.2 Member Enumeration Documentation

##### 3.19.2.1 enum Opcodes

Contains symbolic names for the opcodes that define CSTA phase 2 operations.

## 3.20 Phase3Opcodes Class Reference

### Public Types

- enum [Opcodes](#)

### 3.20.1 Detailed Description

This class contains a public enum that contains symbolic names for the opcodes that define CSTA phase 3 operations.

### 3.20.2 Member Enumeration Documentation

#### 3.20.2.1 enum Opcodes

Contains symbolic names for the opcodes that define CSTA phase 3 operations.

## **3.21 ROSEParseInfo Class Reference**

### **3.21.1 Detailed Description**

Contains information about a received CSTA message obtained by decoding the message's ROSE header.

## 3.22 SiemensCap Class Reference

Inherits [Com::Objsys::Csta::Phase3::IETF\\_CSTAp3](#).

Inherited by [SiemensHipath4000](#).

### 3.22.1 Detailed Description

Implements CSTA phase 3 operations for the Siemens CAP PBX device.



## 3.23 SiemensHicom300 Class Reference

Inherits [Com::Objsys::Csta::Phase1::GenericCSTAp1](#).

### Public Member Functions

- [SiemensHicom300 \(\)](#)

#### 3.23.1 Detailed Description

Implements CSTA phase 1 operations for the Siemens Hicom 300 PBX device.

#### 3.23.2 Constructor & Destructor Documentation

##### 3.23.2.1 SiemensHicom300 ()

Default constructor. This device uses IETF-like prefixes plus a special application code.

## 3.24 SiemensHipath3000p2 Class Reference

Inherits [Com::Objsys::Csta::Phase2::GenericCSTAp2](#).

### Public Member Functions

- [SiemensHipath3000p2 \(\)](#)

#### 3.24.1 Detailed Description

Implements CSTA phase 2 operations for the Siemens Hipath 3000 PBX device.

#### 3.24.2 Constructor & Destructor Documentation

##### 3.24.2.1 SiemensHipath3000p2 ()

Default constructor. This device uses IETF-like prefixes plus a special application code.

## 3.25 SiemensHipath3000p3 Class Reference

Inherits [Com::Objsys::Csta::Phase3::GenericCSTAp3](#).

Inherited by [UnifyOpenscapeX5](#).

### Public Member Functions

- [SiemensHipath3000p3 \(\)](#)

#### 3.25.1 Detailed Description

Implements CSTA phase 3 operations for the Siemens Hipath 3000 PBX device.

#### 3.25.2 Constructor & Destructor Documentation

##### 3.25.2.1 SiemensHipath3000p3 ()

Default constructor. This device uses IETF-like prefixes plus a special application code.

## **3.26 SiemensHipath4000 Class Reference**

Inherits [Com::Objsys::Csta::Devices::SiemensCap](#).

### **3.26.1 Detailed Description**

Implements CSTA phase 3 operations for the Siemens Hipath 4000 PBX device.

## **3.27 SiemensRealitis Class Reference**

Inherits [Com::Objsys::Csta::Phase1::IETF\\_CSTAp1](#).

### **3.27.1 Detailed Description**

Implements CSTA phase 1 operations for the Siemens Realitis PBX device.

## 3.28 SingleStepTransferInfo Class Reference

### Properties

- string [TransferFromDevice](#) [get, set]
- string [TransferringCallID](#) [get, set]
- string [TransferToDevice](#) [get, set]

### 3.28.1 Detailed Description

Contains information needed to complete a phase 3 single step transfer request.

### 3.28.2 Property Documentation

#### 3.28.2.1 string TransferFromDevice [get, set]

Identification (e.g., phone number) of the device from which the call is being transferred.

#### 3.28.2.2 string TransferringCallID [get, set]

The call id number associated with the device from which the call is being transferred.

#### 3.28.2.3 string TransferToDevice [get, set]

Identification (e.g., phone number) of the device to which the call is being transferred.

## 3.29 SocketState Class Reference

### Properties

- byte[] [AckBuffer](#) [get, set]
- byte[] [ReadBuffer](#) [get, set]
- List< byte[] > [ReadBuffers](#) [get, set]
- int [TotalLength](#) [get, set]

### 3.29.1 Detailed Description

This class contains the response received from the PBX and state information about the exchange with the PBX that is used internally by CSTADLL.

### 3.29.2 Property Documentation

#### 3.29.2.1 byte [] AckBuffer [get, set]

Contains the first response from the PBX for situations where the PBX sends multiple response messages (e.g., Get Switching Function [Devices](#)). The data messages that are sent after this ack will be in ReadBuffers.

#### 3.29.2.2 byte [] ReadBuffer [get, set]

Contains the bytes most recently read from the socket. This buffer will be filled in bit by bit as the message is read.

#### 3.29.2.3 List<byte[]> ReadBuffers [get, set]

Contains multiple collections of bytes read from the socket. This array is used for situations where a response to a message comes in multiple segments (e.g., Get Switching Function [Devices](#)). For these situations the immediate response will be in AckBuffer.

#### 3.29.2.4 int TotalLength [get, set]

The total length of a complete message received from the PBX. This is also used as an offset into the read buffer so we can build the message as it's received.

## **3.30 TadiranCoral Class Reference**

Inherits [Com::Objsys::Csta::Phase1::IETF\\_CSTAp1](#).

### **3.30.1 Detailed Description**

Implements CSTA phase 1 operations for the Tadiran Coral PBX device.



## 3.31 TransferCallInfo Class Reference

### Properties

- string [ActiveDeviceCallID](#) [get, set]
- string [ActiveDeviceNumber](#) [get, set]
- string [ConnectedDevice](#) [get, set]
- string [HeldDevice](#) [get, set]
- string [HeldDeviceCallID](#) [get, set]
- string [HeldDeviceNumber](#) [get, set]

### 3.31.1 Detailed Description

Contains information needed to complete a phase 3 transfer call request.

### 3.31.2 Property Documentation

#### 3.31.2.1 string ActiveDeviceCallID [get, set]

The call id associated with the device to which the call is being transferred.

#### 3.31.2.2 string ActiveDeviceNumber [get, set]

The phone number to which the call is being transferred. This number is not necessarily the same as the value for ConnectedDevice.

#### 3.31.2.3 string ConnectedDevice [get, set]

Identification (e.g., phone number) of the device to which the call is being transferred.

#### 3.31.2.4 string HeldDevice [get, set]

Identification (e.g., phone number) of the device from which the call is being transferred.

#### 3.31.2.5 string HeldDeviceCallID [get, set]

The call id associated with the device from which the call is being transferred.

#### 3.31.2.6 string HeldDeviceNumber [get, set]

The phone number from which the call is being transferred. This number is not necessarily the same as the value for HeldDevice.

## 3.32 UnifyOpenscapeX5 Class Reference

Inherits [Com::Objsys::Csta::Devices::SiemensHipath3000p3](#).

### Public Member Functions

- [UnifyOpenscapeX5 \(\)](#)

#### 3.32.1 Detailed Description

Implements CSTA phase 3 operations for the Unify Openscape Business X5 PBX device.

#### 3.32.2 Constructor & Destructor Documentation

##### 3.32.2.1 UnifyOpenscapeX5 ()

Default constructor. This device uses IETF-like prefixes plus a special application code.

# Index

- AckBuffer
  - Com::Objsys::Csta::Common::SocketState, 59
- AcquireControlRight
  - Com::Objsys::Csta::Devices::PanasonicKXTDE, 39
- ActiveDeviceCallID
  - Com::Objsys::Csta::Phase3::TransferCallInfo, 61
- ActiveDeviceNumber
  - Com::Objsys::Csta::Phase3::TransferCallInfo, 61
- Alcatel4400
  - Com::Objsys::Csta::Devices::Alcatel4400, 9
- AlcatelOXO
  - Com::Objsys::Csta::Devices::AlcatelOXO, 10
- AlcatelOXOp1
  - Com::Objsys::Csta::Devices::AlcatelOXOp1, 11
- AnswerCall
  - Com::Objsys::Csta::Phase1::GenericCSTAp1, 14, 15
  - Com::Objsys::Csta::Phase2::GenericCSTAp2, 20, 21
  - Com::Objsys::Csta::Phase3::GenericCSTAp3, 27
- AsyncCallback
  - Com::Objsys::Csta::Common::PBXSessionHelper, 44
- ClearConnection
  - Com::Objsys::Csta::Phase1::GenericCSTAp1, 15
  - Com::Objsys::Csta::Phase2::GenericCSTAp2, 21
  - Com::Objsys::Csta::Phase3::GenericCSTAp3, 27
- ClearMessageWaiting
  - Com::Objsys::Csta::Devices::PanasonicKXTDE, 39
  - Com::Objsys::Csta::Phase3::GenericCSTAp3, 27
- ClientCallback
  - Com::Objsys::Csta::Common::PBXSessionHelper, 46
- Close
  - Com::Objsys::Csta::Common::PBXSessionHelper, 44
- Com.Objsys.Csta.Common, 3
- Com.Objsys.Csta.Devices, 4
- Com.Objsys.Csta.Phase1, 5
- Com.Objsys.Csta.Phase2, 6
- Com.Objsys.Csta.Phase3, 7
- Com::Objsys::Csta::Common::CSTAContext, 12
  - ResponseFromPBX, 12
  - ResponsesFromPBX, 12
- Com::Objsys::Csta::Common::CSTAResponseInfo, 13
  - ResponseFromPBX, 13
  - ResponsesFromPBX, 13
  - StatusCode, 13
  - StatusMessage, 13
- Com::Objsys::Csta::Common::LicenseException, 37
- Com::Objsys::Csta::Common::PBXSessionException, 43
- Com::Objsys::Csta::Common::PBXSessionHelper, 44
  - AsyncCallback, 44
  - ClientCallback, 46
  - Close, 44
  - ConnectionCallback, 44
  - ConnectionLostCallback, 46
  - DebugMode, 46
  - Init, 45
  - LoggingEnabled, 46
  - LoggingFolder, 46
  - Open, 45
  - SendACSEMessage, 45
  - SendMessage, 45, 46
  - WaitForROSEResponse, 46
- Com::Objsys::Csta::Common::ROSEParseInfo, 51
- Com::Objsys::Csta::Common::SocketState, 59
  - AckBuffer, 59
  - ReadBuffer, 59
  - ReadBuffers, 59
  - TotalLength, 59
- Com::Objsys::Csta::Devices::Alcatel4400, 9
  - Alcatel4400, 9
  - MakeACSEAssociation, 9
- Com::Objsys::Csta::Devices::AlcatelOXO, 10
  - AlcatelOXO, 10
  - MakeACSEAssociation, 10
- Com::Objsys::Csta::Devices::AlcatelOXOp1, 11
  - AlcatelOXOp1, 11
  - MakeACSEAssociation, 11
- Com::Objsys::Csta::Devices::PanasonicKXTDA, 38
- Com::Objsys::Csta::Devices::PanasonicKXTDE, 39
  - AcquireControlRight, 39
  - ClearMessageWaiting, 39
  - GetGroupMembers, 39
  - GetSFDevices, 40
  - PDFStart, 40
  - PDFStop, 40
  - ReleaseControlRight, 40

- ResetDisplay, 41
- SendKmeMessage, 41
- SetMessageWaiting, 41
- Com::Objsys::Csta::Devices::PanasonicNCP, 42
- Com::Objsys::Csta::Devices::SiemensCap, 52
- Com::Objsys::Csta::Devices::SiemensHicom300, 53
  - SiemensHicom300, 53
- Com::Objsys::Csta::Devices::SiemensHipath3000p2, 54
  - SiemensHipath3000p2, 54
- Com::Objsys::Csta::Devices::SiemensHipath3000p3, 55
  - SiemensHipath3000p3, 55
- Com::Objsys::Csta::Devices::SiemensHipath4000, 56
- Com::Objsys::Csta::Devices::SiemensRealitis, 57
- Com::Objsys::Csta::Devices::TadiranCoral, 60
- Com::Objsys::Csta::Devices::UnifyOpenscapeX5, 62
  - UnifyOpenscapeX5, 62
- Com::Objsys::Csta::Phase1::GenericCSTAp1, 14
  - AnswerCall, 14, 15
  - ClearConnection, 15
  - ConsultationCall, 15
  - DivertCall, 15
  - EncodeROSERequestHeader, 16
  - GenericCSTAp1, 14
  - MakeACSEAssociation, 16
  - MakeCall, 16
  - MonitorStart, 16
  - MonitorStop, 17
  - QueryDevice, 17
  - ReleaseACSEAssociation, 18
  - SnapshotDevice, 18
  - ThreadContext, 19
  - TransferCall, 18
- Com::Objsys::Csta::Phase1::IETF\_CSTAp1, 34
  - IETF\_CSTAp1, 34
- Com::Objsys::Csta::Phase1::Phase1Opcodes, 48
  - Opcodes, 48
- Com::Objsys::Csta::Phase2::GenericCSTAp2, 20
  - AnswerCall, 20, 21
  - ClearConnection, 21
  - ConsultationCall, 21
  - DivertCall, 21
  - EncodeROSERequestHeader, 22
  - GenericCSTAp2, 20
  - MakeACSEAssociation, 22
  - MakeCall, 22
  - MonitorStart, 22
  - MonitorStop, 23
  - QueryDevice, 23
  - ReleaseACSEAssociation, 24
  - SnapshotDevice, 24
  - ThreadContext, 25
  - TransferCall, 24
- Com::Objsys::Csta::Phase2::IETF\_CSTAp2, 35
  - IETF\_CSTAp2, 35
- Com::Objsys::Csta::Phase2::Phase2Opcodes, 49
  - Opcodes, 49
- Com::Objsys::Csta::Phase3::GenericCSTAp3, 26
  - AnswerCall, 27
  - ClearConnection, 27
  - ClearMessageWaiting, 27
  - ConsultationCall, 28
  - EncodeROSERequestHeader, 28
  - GenericCSTAp3, 26
  - GetSFDDevices, 28
  - MakeACSEAssociation, 29
  - MakeCall, 29
  - MonitorStart, 29
  - MonitorStop, 29, 30
  - ReleaseACSEAssociation, 30
  - RingDevice, 30
  - SendData, 30
  - SetDisplay, 31
  - SetMessageWaiting, 31
  - SingleStepTransfer, 31
  - SnapshotDevice, 32
  - StartDataPath, 32
  - StopDataPath, 32
  - StopRing, 32
  - ThreadContext, 33
  - TransferCall, 33
- Com::Objsys::Csta::Phase3::IETF\_CSTAp3, 36
  - IETF\_CSTAp3, 36
- Com::Objsys::Csta::Phase3::Phase3Opcodes, 50
  - Opcodes, 50
- Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 58
  - TransferFromDevice, 58
  - TransferringCallID, 58
  - TransferToDevice, 58
- Com::Objsys::Csta::Phase3::TransferCallInfo, 61
  - ActiveDeviceCallID, 61
  - ActiveDeviceNumber, 61
  - ConnectedDevice, 61
  - HeldDevice, 61
  - HeldDeviceCallID, 61
  - HeldDeviceNumber, 61
- ConnectedDevice
  - Com::Objsys::Csta::Phase3::TransferCallInfo, 61
- ConnectionCallback
  - Com::Objsys::Csta::Common::PBXSessionHelper, 44
- ConnectionLostCallback
  - Com::Objsys::Csta::Common::PBXSessionHelper, 46
- ConsultationCall
  - Com::Objsys::Csta::Phase1::GenericCSTAp1, 15
  - Com::Objsys::Csta::Phase2::GenericCSTAp2, 21
  - Com::Objsys::Csta::Phase3::GenericCSTAp3, 28

DebugMode  
     Com::Objsys::Csta::Common::PBXSessionHelper, 46

DivertCall  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 15  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 21

EncodeROSErequestHeader  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 16  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 22  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 28

GenericCSTAp1  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 14

GenericCSTAp2  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 20

GenericCSTAp3  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 26

GetGroupMembers  
     Com::Objsys::Csta::Devices::PanasonicKXTDE, 39

GetSFDevices  
     Com::Objsys::Csta::Devices::PanasonicKXTDE, 40  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 28

HeldDevice  
     Com::Objsys::Csta::Phase3::TransferCallInfo, 61

HeldDeviceCallID  
     Com::Objsys::Csta::Phase3::TransferCallInfo, 61

HeldDeviceNumber  
     Com::Objsys::Csta::Phase3::TransferCallInfo, 61

IETF\_CSTAp1  
     Com::Objsys::Csta::Phase1::IETF\_CSTAp1, 34

IETF\_CSTAp2  
     Com::Objsys::Csta::Phase2::IETF\_CSTAp2, 35

IETF\_CSTAp3  
     Com::Objsys::Csta::Phase3::IETF\_CSTAp3, 36

Init  
     Com::Objsys::Csta::Common::PBXSessionHelper, 45

LoggingEnabled  
     Com::Objsys::Csta::Common::PBXSessionHelper, 46

LoggingFolder  
     Com::Objsys::Csta::Common::PBXSessionHelper, 46

MakeACSEAssociation  
     Com::Objsys::Csta::Devices::Alcatel4400, 9  
     Com::Objsys::Csta::Devices::AlcatelOXO, 10  
     Com::Objsys::Csta::Devices::AlcatelOXOp1, 11  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 16  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 22  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 29

MakeCall  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 16  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 22  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 29

MonitorStart  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 16  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 22  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 29

MonitorStop  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 17  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 23  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 29, 30

Opcodes  
     Com::Objsys::Csta::Phase1::Phase1Opcodes, 48  
     Com::Objsys::Csta::Phase2::Phase2Opcodes, 49  
     Com::Objsys::Csta::Phase3::Phase3Opcodes, 50

Open  
     Com::Objsys::Csta::Common::PBXSessionHelper, 45

PDFStart  
     Com::Objsys::Csta::Devices::PanasonicKXTDE, 40

PDFStop  
     Com::Objsys::Csta::Devices::PanasonicKXTDE, 40

QueryDevice  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 17  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 23

ReadBuffer  
     Com::Objsys::Csta::Common::SocketState, 59

ReadBuffers  
     Com::Objsys::Csta::Common::SocketState, 59

ReleaseACSEAssociation  
     Com::Objsys::Csta::Phase1::GenericCSTAp1, 18  
     Com::Objsys::Csta::Phase2::GenericCSTAp2, 24  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 30

ReleaseControlRight  
     Com::Objsys::Csta::Devices::PanasonicKXTDE, 40

ResetDisplay  
     Com::Objsys::Csta::Devices::PanasonicKXTDE, 41

ResponseFromPBX  
     Com::Objsys::Csta::Common::CSTAContext, 12  
     Com::Objsys::Csta::Common::CSTARResponseInfo, 13

ResponsesFromPBX  
     Com::Objsys::Csta::Common::CSTAContext, 12  
     Com::Objsys::Csta::Common::CSTARResponseInfo, 13

RingDevice  
     Com::Objsys::Csta::Phase3::GenericCSTAp3, 30

SendACSEMessage

Com::Objsys::Csta::Common::PBXSessionHelper, 45  
 SendData  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 30  
 SendKmeMessage  
   Com::Objsys::Csta::Devices::PanasonicKXTDE, 41  
 SendMessage  
   Com::Objsys::Csta::Common::PBXSessionHelper, 45, 46  
 SetDisplay  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 31  
 SetMessageWaiting  
   Com::Objsys::Csta::Devices::PanasonicKXTDE, 41  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 31  
 SiemensHicom300  
   Com::Objsys::Csta::Devices::SiemensHicom300, 53  
 SiemensHipath3000p2  
   Com::Objsys::Csta::Devices::SiemensHipath3000p2, 54  
 SiemensHipath3000p3  
   Com::Objsys::Csta::Devices::SiemensHipath3000p3, 55  
 SingleStepTransfer  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 31  
 SnapshotDevice  
   Com::Objsys::Csta::Phase1::GenericCSTAp1, 18  
   Com::Objsys::Csta::Phase2::GenericCSTAp2, 24  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 32  
 StartDataPath  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 32  
 StatusCode  
   Com::Objsys::Csta::Common::CSTAResponseInfo, 13  
 StatusMessage  
   Com::Objsys::Csta::Common::CSTAResponseInfo, 13  
 StopDataPath  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 32  
 StopRing  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 32  
  
 ThreadContext  
   Com::Objsys::Csta::Phase1::GenericCSTAp1, 19  
   Com::Objsys::Csta::Phase2::GenericCSTAp2, 25  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 33  
 TotalLength  
   Com::Objsys::Csta::Common::SocketState, 59  
 TransferCall  
   Com::Objsys::Csta::Phase1::GenericCSTAp1, 18  
   Com::Objsys::Csta::Phase2::GenericCSTAp2, 24  
   Com::Objsys::Csta::Phase3::GenericCSTAp3, 33  
 TransferFromDevice  
   Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 58  
  
 TransferringCallID  
   Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 58  
 TransferToDevice  
   Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 58  
 UnifyOpenscapeX5  
   Com::Objsys::Csta::Devices::UnifyOpenscapeX5, 62  
 WaitForROSEResponse  
   Com::Objsys::Csta::Common::PBXSessionHelper, 46