

CSTA

CSTADLL
Version 2.3.4
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 ©1997-2017 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.

Contents

1	CSTABERPhase3	1
2	Namespace Documentation	3
2.1	Package Com.Objsys.Csta.Devices	3
2.1.1	Detailed Description	3
2.2	Package Com.Objsys.Csta.Phase3	4
2.2.1	Detailed Description	4
3	Class Documentation	5
3.1	Constants Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Enumeration Documentation	5
3.1.2.1	ACSEMessageTypes	5
3.1.2.2	CallbackInvocationMechanisms	5
3.1.2.3	CommunicationTypes	6
3.1.2.4	Encoding	6
3.1.2.5	PBXModels	6
3.1.2.6	XMLSessionMessageTypes	6
3.1.3	Member Data Documentation	6
3.1.3.1	MAX_LOGFILE_SIZE	6
3.2	CSTAContext Class Reference	7
3.2.1	Detailed Description	7
3.2.2	Property Documentation	7
3.2.2.1	ResponseFromPBX	7
3.2.2.2	ResponsesFromPBX	7
3.2.2.3	XMLResponseFromPBX	7
3.2.2.4	XMLResponsesFromPBX	7
3.3	CSTAResponseInfo Class Reference	8
3.3.1	Detailed Description	8

3.3.2	Property Documentation	8
3.3.2.1	ResponseFromPBX	8
3.3.2.2	ResponsesFromPBX	8
3.3.2.3	ResponsesFromUA	8
3.3.2.4	StatusCode	9
3.3.2.5	StatusMessage	9
3.3.2.6	XMLResponseFromPBX	9
3.3.2.7	XMLResponseFromUA	9
3.3.2.8	XMLResponsesFromPBX	9
3.4	GenericCSTap3 Class Reference	10
3.4.1	Detailed Description	12
3.4.2	Constructor & Destructor Documentation	13
3.4.2.1	GenericCSTap3	13
3.4.2.2	GenericCSTap3	13
3.4.3	Member Function Documentation	13
3.4.3.1	AlternateCall	13
3.4.3.2	AnswerCall	13
3.4.3.3	AnswerCall	14
3.4.3.4	AnswerCall	14
3.4.3.5	ClearConnection	14
3.4.3.6	ClearDoNotDisturb	14
3.4.3.7	ClearMessageWaiting	14
3.4.3.8	ConferenceCall	15
3.4.3.9	ConsultationCall	15
3.4.3.10	EncodeACSEConnectionRequest	15
3.4.3.11	EncodeACSEReleaseRequest	16
3.4.3.12	EncodeAlternateCallRequest	16
3.4.3.13	EncodeAnswerCallRequest	16
3.4.3.14	EncodeAnswerCallRequest	17
3.4.3.15	EncodeAnswerCallRequest	17
3.4.3.16	EncodeClearConnectionRequest	17
3.4.3.17	EncodeConferenceCallRequest	17
3.4.3.18	EncodeConsultationCallRequest	18
3.4.3.19	EncodeGetAgentStateRequest	18
3.4.3.20	EncodeGetDNDRRequest	18
3.4.3.21	EncodeGetLogicalDevInfoRequest	19
3.4.3.22	EncodeGetPhysicalDevInfoRequest	19

3.4.3.23	EncodeGetSFDevicesRequest	19
3.4.3.24	EncodeHoldCallRequest	20
3.4.3.25	EncodeMakeCallRequest	20
3.4.3.26	EncodeMonitorStartRequest	20
3.4.3.27	EncodeMonitorStartRequest	21
3.4.3.28	EncodeMonitorStartRequest	21
3.4.3.29	EncodeMonitorStopRequest	21
3.4.3.30	EncodeMonitorStopRequest	21
3.4.3.31	EncodeRequestSysStatRequest	22
3.4.3.32	EncodeRetrieveCallRequest	22
3.4.3.33	EncodeROSERequestHeader	22
3.4.3.34	EncodeSendDataRequest	23
3.4.3.35	EncodeSendStoredCDRRequest	23
3.4.3.36	EncodeSetAgentStateRequest	23
3.4.3.37	EncodeSetAgentStateRequest	24
3.4.3.38	EncodeSetDisplayRequest	24
3.4.3.39	EncodeSetOrClearDNDRequest	24
3.4.3.40	EncodeSetOrClearMWRequest	25
3.4.3.41	EncodeSetRingerStatusRequest	25
3.4.3.42	EncodeSingleStepTransferRequest	25
3.4.3.43	EncodeSingleStepTransferRequest	26
3.4.3.44	EncodeSnapshotCallRequest	26
3.4.3.45	EncodeSnapshotDeviceRequest	26
3.4.3.46	EncodeStartCDRTransRequest	27
3.4.3.47	EncodeStartDataPathRequest	27
3.4.3.48	EncodeStopCDRTransRequest	27
3.4.3.49	EncodeStopDataPathRequest	27
3.4.3.50	EncodeTransferCallRequest	28
3.4.3.51	EncodeTransferCallRequest	28
3.4.3.52	GetAgentState	28
3.4.3.53	GetDoNotDisturb	29
3.4.3.54	GetLogicalDevInfo	29
3.4.3.55	GetPhysicalDevInfo	29
3.4.3.56	GetSFDevices	29
3.4.3.57	GetSFDevices	30
3.4.3.58	HoldCall	30
3.4.3.59	MakeACSEAssociation	30

3.4.3.60	MakeCall	30
3.4.3.61	MonitorStart	30
3.4.3.62	MonitorStart	31
3.4.3.63	MonitorStart	31
3.4.3.64	MonitorStop	31
3.4.3.65	MonitorStop	31
3.4.3.66	MonitorStop	32
3.4.3.67	MonitorStopAtNumericDevice	32
3.4.3.68	ReleaseACSEAssociation	32
3.4.3.69	RequestSystemStatus	32
3.4.3.70	RetrieveCall	32
3.4.3.71	RingDevice	33
3.4.3.72	SendData	33
3.4.3.73	SendStoredCDR	33
3.4.3.74	SetAgentState	33
3.4.3.75	SetAgentState	34
3.4.3.76	SetDisplay	34
3.4.3.77	SetDoNotDisturb	34
3.4.3.78	SetMessageWaiting	34
3.4.3.79	SingleStepTransfer	35
3.4.3.80	SingleStepTransfer	35
3.4.3.81	SnapshotCall	35
3.4.3.82	SnapshotDevice	35
3.4.3.83	StartCDRTransmission	36
3.4.3.84	StartDataPath	36
3.4.3.85	StopCDRTransmission	36
3.4.3.86	StopDataPath	36
3.4.3.87	StopRing	37
3.4.3.88	TransferCall	37
3.4.3.89	TransferCall	37
3.4.4	Property Documentation	37
3.4.4.1	SessionObject	37
3.4.4.2	ThreadContext	37
3.5	IETF_CSTAp3 Class Reference	38
3.5.1	Detailed Description	38
3.5.2	Constructor & Destructor Documentation	38
3.5.2.1	IETF_CSTAp3	38

3.5.2.2	IETF_CSTAp3	38
3.6	LicenseException Class Reference	39
3.6.1	Detailed Description	39
3.7	LicenseOptions Class Reference	40
3.7.1	Detailed Description	40
3.8	PanasonicKXTDA Class Reference	41
3.8.1	Detailed Description	41
3.8.2	Constructor & Destructor Documentation	41
3.8.2.1	PanasonicKXTDA	41
3.8.2.2	PanasonicKXTDA	41
3.9	PanasonicKXTDE Class Reference	42
3.9.1	Detailed Description	42
3.9.2	Member Enumeration Documentation	42
3.9.2.1	DeviceDataTypes	42
3.9.3	Constructor & Destructor Documentation	43
3.9.3.1	PanasonicKXTDE	43
3.9.3.2	PanasonicKXTDE	43
3.9.4	Member Function Documentation	43
3.9.4.1	AcquireControlRight	43
3.9.4.2	ClearMessageWaiting	43
3.9.4.3	EncodeACSEConnectionRequest	43
3.9.4.4	EncodeMakeCallRequest	44
3.9.4.5	EncodeSetDisplayRequest	44
3.9.4.6	GetDeviceData	44
3.9.4.7	GetGroupMembers	45
3.9.4.8	GetSFDevices	45
3.9.4.9	PDFStart	45
3.9.4.10	PDFStop	45
3.9.4.11	ReleaseControlRight	46
3.9.4.12	ResetDisplay	46
3.9.4.13	SendKmeMessage	46
3.9.4.14	SetMessageWaiting	46
3.10	PanasonicNCP Class Reference	47
3.10.1	Detailed Description	47
3.10.2	Constructor & Destructor Documentation	47
3.10.2.1	PanasonicNCP	47
3.10.2.2	PanasonicNCP	47

3.11 PBXSession Class Reference	48
3.11.1 Detailed Description	48
3.11.2 Constructor & Destructor Documentation	49
3.11.2.1 PBXSession	49
3.11.3 Member Function Documentation	49
3.11.3.1 AsyncCallback	49
3.11.3.2 AsyncExceptionCallback	49
3.11.3.3 Close	49
3.11.3.4 ConnectionCallback	49
3.11.3.5 Open	50
3.11.3.6 SendACSEMessage	50
3.11.3.7 SendMessage	50
3.11.3.8 SendMessage	50
3.11.3.9 SendXMLMessage	51
3.11.3.10 SendXMLMessage	51
3.11.3.11 SendXMLSession	51
3.11.3.12 WaitForROSEResponse	51
3.11.3.13 WaitForXMLResponse	51
3.11.3.14 XMLAsyncCallback	52
3.11.4 Property Documentation	52
3.11.4.1 CallbackInvocationMechanism	52
3.11.4.2 CDRCallback	52
3.11.4.3 ClientCallback	52
3.11.4.4 Connected	52
3.11.4.5 ConnectionLostCallback	52
3.11.4.6 DebugMode	52
3.11.4.7 ExceptionCallback	52
3.11.4.8 MaxReceiveTimeout	53
3.11.4.9 MessageEncoding	53
3.11.4.10 PBXSystem	53
3.11.4.11 Port	53
3.11.4.12 SystemStatusCallback	53
3.11.4.13 XMLCDRCallback	53
3.11.4.14 XMLClientCallback	53
3.11.4.15 XMLSystemStatusCallback	53
3.12 PBXSessionException Class Reference	54
3.12.1 Detailed Description	54

3.13 PBXSessionHelper Class Reference	55
3.13.1 Detailed Description	55
3.13.2 Property Documentation	55
3.13.2.1 LoggingEnabled	55
3.13.2.2 LoggingFolder	55
3.13.2.3 MaxLogFileSize	55
3.14 PBXSessionHelperPhase3 Class Reference	56
3.14.1 Detailed Description	56
3.15 Phase3OpCodes Class Reference	57
3.15.1 Detailed Description	57
3.15.2 Member Enumeration Documentation	57
3.15.2.1 OpCodes	57
3.16 ROSEParseInfo Class Reference	58
3.16.1 Detailed Description	58
3.17 SiemensCap Class Reference	59
3.17.1 Detailed Description	59
3.17.2 Constructor & Destructor Documentation	59
3.17.2.1 SiemensCap	59
3.17.2.2 SiemensCap	59
3.17.3 Member Function Documentation	59
3.17.3.1 EncodeACSEConnectionRequest	59
3.17.3.2 MakeACSEAssociation	60
3.17.3.3 MakeACSEAssociation	60
3.17.3.4 MakeACSEAssociation	60
3.18 SiemensHipath3000p3 Class Reference	62
3.18.1 Detailed Description	62
3.18.2 Constructor & Destructor Documentation	62
3.18.2.1 SiemensHipath3000p3	62
3.18.2.2 SiemensHipath3000p3	62
3.18.3 Member Function Documentation	62
3.18.3.1 EncodeACSEConnectionRequest	62
3.18.3.2 EncodeSingleStepTransferRequest	63
3.18.3.3 MakeACSEAssociation	63
3.19 SiemensHipath4000 Class Reference	64
3.19.1 Detailed Description	64
3.19.2 Constructor & Destructor Documentation	64
3.19.2.1 SiemensHipath4000	64

3.19.2.2	SiemensHipath4000	64
3.20	SingleStepTransferInfo Class Reference	65
3.20.1	Detailed Description	65
3.20.2	Property Documentation	65
3.20.2.1	TransferFromDevice	65
3.20.2.2	TransferringCallID	65
3.20.2.3	TransferToDevice	65
3.21	SocketState Class Reference	66
3.21.1	Detailed Description	66
3.21.2	Property Documentation	66
3.21.2.1	AckBuffer	66
3.21.2.2	ReadBuffer	66
3.21.2.3	ReadBuffers	66
3.21.2.4	TotalLength	66
3.22	TransferCallInfo Class Reference	67
3.22.1	Detailed Description	67
3.22.2	Property Documentation	67
3.22.2.1	ActiveDeviceCallID	67
3.22.2.2	ActiveDeviceNumber	67
3.22.2.3	ConnectedDevice	67
3.22.2.4	HeldDevice	67
3.22.2.5	HeldDeviceCallID	67
3.22.2.6	HeldDeviceNumber	67
3.23	UnifyOpenscape4000BER Class Reference	68
3.23.1	Detailed Description	68
3.23.2	Constructor & Destructor Documentation	68
3.23.2.1	UnifyOpenscape4000BER	68
3.23.2.2	UnifyOpenscape4000BER	68
3.23.3	Member Function Documentation	68
3.23.3.1	EncodeACSEConnectionRequest	68
3.23.3.2	MakeACSEAssociation	69
3.24	UnifyOpenscapeX5 Class Reference	70
3.24.1	Detailed Description	70
3.24.2	Constructor & Destructor Documentation	70
3.24.2.1	UnifyOpenscapeX5	70
3.24.2.2	UnifyOpenscapeX5	70

Chapter 1

CSTABERPhase3

CSTABERPhase3 is a Microsoft .NET 4.5 DLL that allows client code to communicate with a PBX device.

The DLL uses the following namespaces:

- `Com.Objsys.Csta.Devices`
- `Com.Objsys.Csta.Phase3`

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.

A typical way to use the DLL is to use the `PBXSession` class to set up the communication to the PBX device via the constructor. If the PBX will be sending asynchronous data, such as monitor packets, to the client, the `ClientCallback` or `XMLClientCallback` 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.

If the PBX will be sending Call Detail Records Report or Call Detail Records Notification messages to the client, the `CDRCallback` or `XMLCDRCallback` property can be used to define a callback method to receive the messages. If no callback method is defined, Call Detail messages will be ignored.

The CSTADLL kit includes some samples to guide you in writing your own code. The samples are evenly split between those implemented in C# and those implemented in Visual BASIC. Each language has samples for communicating with PBX devices that use BER CSTA and with PBX devices that use XML CSTA.

The classes and methods exposed by the DLL are probably sufficient to handle operations for most PBX devices. But if needed, you can write a class of your own to handle operations for a PBX device that the software doesn't explicitly support. The sample `NewPBX` 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. The assumption in the sample is that this device uses standard messages for all operations except for the initial association messages. These messages are the ones that are most commonly different from one PBX to the next. The `NewPBX` sample shows how the `EncodeACSEConnectionRequest()` method within the `GenericCSTAp2` class (for BER PBX devices) or the `EncodeStartSession()` method within the `GenericXMLED4` class (for XML PBX devices) can be overridden in a class that you can write. The override implementation handles the details that are specific to the device.

The 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. The default behavior is that 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. That default size of 5 Mb can be modified by using the `MaxLogFileSize` property of the `PBXSessionHelper` class.

If your CSTADLL kit is licensed (i.e., not unlimited), then you will need to deploy your application with the DLLs `Reprise.dll` and `rlm1212.dll` that are in the kit. The file `rlm1212.dll` is a 32-bit native DLL as opposed to a .NET DLL. As such, if you build your code with a Makefile, you will need to use the `/platform:x86` qualifier to the `csc` or `vbc` command. If you build your code with a Visual Studio project, you will need to use `x86` as the target platform instead of `AnyCPU`. These steps are to ensure proper interfacing to the native 32-bit `rlm1212.dll`. There is also a 64-bit version of `rlm1212.dll` available if you prefer to target the x64 platform.

Chapter 2

Namespace Documentation

2.1 Package `Com.Objsys.Csta.Devices`

Classes

- class [PanasonicKXTDA](#)
- class [PanasonicKXTDE](#)
- class [PanasonicNCP](#)
- class [SiemensCap](#)
- class [SiemensHipath3000p3](#)
- class [SiemensHipath4000](#)
- class [UnifyOpenscape4000BER](#)
- class [UnifyOpenscapeX5](#)

2.1.1 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.

2.2 Package Com.Objsys.Csta.Phase3

Classes

- class [Constants](#)
- class [CSTAContext](#)
- class [CSTAResponseInfo](#)
- class [GenericCSTAp3](#)
- class [IETF_CSTAp3](#)
- class [LicenseException](#)
- class [LicenseOptions](#)
- class [PBXSession](#)
- class [PBXSessionException](#)
- class [PBXSessionHelper](#)
- class [PBXSessionHelperPhase3](#)
- class [Phase3Opcodes](#)
- class [ROSEParseInfo](#)
- class [SingleStepTransferInfo](#)
- class [SocketState](#)
- class [TransferCallInfo](#)

2.2.1 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 Constants Class Reference

Public Types

- enum [ACSEMessageTypes](#)
- enum [CallbackInvocationMechanisms](#)
- enum [CommunicationTypes](#)
- enum [Encoding](#)
- enum [PBXModels](#)
- enum [XMLSessionMessageTypes](#)

Public Attributes

- const long [MAX_LOGFILE_SIZE](#) = (5 * 1024 * 1024)

3.1.1 Detailed Description

The [Constants](#) class contains some helpful constant and enum definitions.

3.1.2 Member Enumeration Documentation

3.1.2.1 enum ACSEMessageTypes

Provides symbolic names for the ACSE message types.

3.1.2.2 enum CallbackInvocationMechanisms

Indicates how an asynchronous callback method should be invoked. This setting influences how the asynchronous callback methods for monitor event report messages, route messages, and Call Detail Record messages are invoked.

The value `InvokeCallbackThenPostNextRead` causes the callback method to be invoked before the next read from the PBX or UA is posted to the socket. This setting is the default. With this mechanism callback methods can be easily debugged because new packets from the PBX or UA won't be arriving while debugging of the method is in progress. This mechanism also ensures that messages from the PBX or UA will arrive in a predictable order.

The value `PostNextReadThenInvokeCallback` causes the callback method to be invoked after the next read from the PBX or UA is posted to the socket. Use of this mechanism is necessary if additional synchronous CSTA messages are going to be sent as part of a callback method's processing. If this mechanism is not used in such a case, the response to the CSTA message sent from the callback method will never be seen because no read to the socket was posted. With that said, however, use this mechanism with EXTREME caution. Because the read to the socket is posted before the event is handled, event n+1 may come in and get handled before event n. You may need to add code to ensure that events get handled in an expected order, if such code is even possible for your situation.

3.1.2.3 enum CommunicationTypes

Provides symbolic names for different ways of communicating with a PBX or UA. The values of this enum influence how each message exchange with a PBX or UA is handled.

3.1.2.4 enum Encoding

Provides symbolic names for the mechanisms for encoding CSTA messages.

3.1.2.5 enum PBXModels

Provides symbolic names for different PBX models.

3.1.2.6 enum XMLSessionMessageTypes

Provides symbolic names for the XML session management message types.

3.1.3 Member Data Documentation

3.1.3.1 const long MAX_LOGFILE_SIZE = (5 * 1024 * 1024)

Defines the maximum size, in bytes, that a log file is allowed to grow to before a new log file is opened.

3.2 CSTAContext Class Reference

Properties

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

3.2.1 Detailed Description

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

3.2.2 Property Documentation

3.2.2.1 `byte [] ResponseFromPBX [get, set]`

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

3.2.2.2 `List<byte[]> ResponsesFromPBX [get, set]`

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

3.2.2.3 `string XMLResponseFromPBX [get, set]`

See documentation for [CSTAResponseInfo.XMLResponseFromPBX](#).

3.2.2.4 `List<string> XMLResponsesFromPBX [get, set]`

See documentation for [CSTAResponseInfo.XMLResponsesFromPBX](#).

3.3 CSTAResponseInfo Class Reference

Properties

- `byte[] ResponseFromPBX` [get, set]
- `List< byte[] > ResponsesFromPBX` [get, set]
- `List< byte[] > ResponsesFromUA` [get, set]
- `int StatusCode` [get, set]
- `string StatusMessage` [get, set]
- `string XMLResponseFromPBX` [get, set]
- `string XMLResponseFromUA` [get, set]
- `List< string > XMLResponsesFromPBX` [get, set]

3.3.1 Detailed Description

Contains information about a PBX operation that was attempted.

3.3.2 Property Documentation

3.3.2.1 `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.

3.3.2.2 `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.

3.3.2.3 `List<byte[]> ResponsesFromUA` [get, set]

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

3.3.2.4 int StatusCode [get, set]

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

3.3.2.5 string StatusMessage [get, set]

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

3.3.2.6 string XMLResponseFromPBX [get, set]

Contains the response from the PBX for messages that generate a single atomic XML response, or the immediate XML acknowledgement response for messages that generate multiple XML data responses (e.g., GetSwitchingFunctionDevices). 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 XMLResponseFromPBX property of the thread's [CSTAContext](#) object. If the value of that property changes, then the value of this property changes.

3.3.2.7 string XMLResponseFromUA [get, set]

Contains the response from the UA for messages that generate a single atomic XML response, or the immediate XML acknowledgement response for messages that generate multiple XML data responses (e.g., GetSwitchingFunctionDevices). If a message that normally generates multiple response segments encounters an error (e.g., the UA rejects the message), then the single error message returned by the UA will be in this property; the ResponsesFromUA property will be null.

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

3.3.2.8 List<string> XMLResponsesFromPBX [get, set]

Contains the responses from the PBX for messages that generate multiple XML response segments (e.g., GetSwitchingFunctionDevices). 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 XMLResponsesFromPBX property of the thread's [CSTAContext](#) object. If the value of that property changes, then the value of this property changes.

3.4 GenericCSTAp3 Class Reference

Inherited by [SiemensHipath3000p3](#), and [IETF_CSTAp3](#).

Public Member Functions

- virtual [CSTARResponseInfo AlternateCall](#) (ConnectionID heldCall, ConnectionID activeCall)
- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer, string deviceToLift)
- virtual [CSTARResponseInfo AnswerCall](#) (ConnectionID callToAnswer)
- virtual [CSTARResponseInfo AnswerCall](#) (string deviceToLift)
- virtual [CSTARResponseInfo ClearConnection](#) (ConnectionID connToClear)
- virtual [CSTARResponseInfo ClearDoNotDisturb](#) (string targetDevice)
- virtual [CSTARResponseInfo ClearMessageWaiting](#) (string targetDevice)
- virtual [CSTARResponseInfo ConferenceCall](#) (ConnectionID heldCall, ConnectionID activeCall)
- virtual [CSTARResponseInfo ConsultationCall](#) (ConnectionID existingCall, string targetDevice)
- virtual int [EncodeROSERequestHeader](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, [Phase3Opcodes.Opcodes](#) opcode)
- [GenericCSTAp3](#) ([PBXSession](#) sessionObject)
- [GenericCSTAp3](#) (string pbxSystem, int port)
- virtual [CSTARResponseInfo GetAgentState](#) (string agentDevice)
- virtual [CSTARResponseInfo GetDoNotDisturb](#) (string targetDevice)
- virtual [CSTARResponseInfo GetLogicalDevInfo](#) (string targetDevice)
- virtual [CSTARResponseInfo GetPhysicalDevInfo](#) (string targetDevice)
- virtual [CSTARResponseInfo GetSFDevices](#) (ReqDeviceCategory deviceCategory)
- virtual [CSTARResponseInfo GetSFDevices](#) ()
- virtual [CSTARResponseInfo HoldCall](#) (ConnectionID callToHold)
- virtual [CSTARResponseInfo MakeACSEAssociation](#) ()
- virtual [CSTARResponseInfo MakeCall](#) (string callingDevice, string calledDevice)
- virtual [CSTARResponseInfo MonitorStart](#) (ConnectionID callToMonitor)
- virtual [CSTARResponseInfo MonitorStart](#) (long deviceToMonitor)
- virtual [CSTARResponseInfo MonitorStart](#) (string deviceToMonitor)
- virtual [CSTARResponseInfo MonitorStop](#) (string monitoredDevice)
- virtual [CSTARResponseInfo MonitorStop](#) (MonitorCrossRefID crossRefID)
- virtual [CSTARResponseInfo MonitorStop](#) (int crossRefInt)
- virtual [CSTARResponseInfo MonitorStopAtNumericDevice](#) (long monitoredDevice)
- virtual [CSTARResponseInfo ReleaseACSEAssociation](#) ()
- virtual [CSTARResponseInfo RequestSystemStatus](#) ()
- virtual [CSTARResponseInfo RetrieveCall](#) (ConnectionID callToRetrieve)
- virtual [CSTARResponseInfo RingDevice](#) (string targetDevice, string targetRinger, long ringPattern)
- virtual [CSTARResponseInfo SendData](#) (IOCrossRefID ioCrossRef, string text)
- virtual [CSTARResponseInfo SendStoredCDR](#) (CDRCrossRefID cdrCrossRefID)
- virtual [CSTARResponseInfo SetAgentState](#) (string agentDevice, ReqAgentState agentState)
- virtual [CSTARResponseInfo SetAgentState](#) (string agentDevice, ReqAgentState agentState, AgentID aid)
- virtual [CSTARResponseInfo SetDisplay](#) (string targetDevice, string text)
- virtual [CSTARResponseInfo SetDoNotDisturb](#) (string targetDevice)
- virtual [CSTARResponseInfo SetMessageWaiting](#) (string targetDevice)
- virtual [CSTARResponseInfo SingleStepTransfer](#) ([SingleStepTransferInfo](#) sstInfo)
- virtual [CSTARResponseInfo SingleStepTransfer](#) (ConnectionID callToTransfer, string transferToDevice)
- virtual [CSTARResponseInfo SnapshotCall](#) (ConnectionID callToSnapshot)
- virtual [CSTARResponseInfo SnapshotDevice](#) (string deviceToSnapshot)

- virtual [CSTARResponseInfo StartCDRTransmission](#) (CDRTransferMode transferMode)
- virtual [CSTARResponseInfo StartDataPath](#) (string targetDevice)
- virtual [CSTARResponseInfo StopCDRTransmission](#) (CDRCrossRefID cdrCrossRefID)
- 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)

Protected Member Functions

- virtual int [EncodeACSEConnectionRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer)
- virtual int [EncodeACSEReleaseRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer)
- virtual int [EncodeAlternateCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID heldCall, ConnectionID activeCall)
- virtual int [EncodeAnswerCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer, string deviceToLift)
- virtual int [EncodeAnswerCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer)
- virtual int [EncodeAnswerCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, string deviceToLift)
- virtual int [EncodeClearConnectionRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID connToClear)
- virtual int [EncodeConferenceCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID heldCall, ConnectionID activeCall)
- virtual int [EncodeConsultationCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID existingCall, string targetDevice)
- virtual int [EncodeGetAgentStateRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, string agentDevice)
- virtual int [EncodeGetDNDRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice)
- virtual int [EncodeGetLogicalDevInfoRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice)
- virtual int [EncodeGetPhysicalDevInfoRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice)
- virtual int [EncodeGetSFDDevicesRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ReqDeviceCategory deviceCategory)
- virtual int [EncodeHoldCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToHold)
- virtual int [EncodeMakeCallRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice)
- virtual int [EncodeMonitorStartRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToMonitor)
- virtual int [EncodeMonitorStartRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, long deviceToMonitor)
- virtual int [EncodeMonitorStartRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, string deviceToMonitor)
- virtual int [EncodeMonitorStopRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, MonitorCrossRefID crossRefObj)
- virtual int [EncodeMonitorStopRequest](#) ([CSTARResponseInfo](#) response, Asn1BerEncodeBuffer encodeBuffer, int crossRefInt, out MonitorCrossRefID crossRefID)

- virtual int [EncodeRequestSysStatRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer)
- virtual int [EncodeRetrieveCallRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToRetrieve)
- virtual int [EncodeSendDataRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, IOCrossRefID ioCrossRef, string text)
- virtual int [EncodeSendStoredCDRRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, CDRCrossRefID cdrCrossRefID)
- virtual int [EncodeSetAgentStateRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string agentDevice, ReqAgentState agentState)
- virtual int [EncodeSetAgentStateRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string agentDevice, ReqAgentState agentState, AgentID aid)
- virtual int [EncodeSetDisplayRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, string text)
- virtual int [EncodeSetOrClearDNDRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool dndOn)
- virtual int [EncodeSetOrClearMWRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool indicatorOn)
- virtual int [EncodeSetRingerStatusRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, string targetRinger, RingMode mode, long ringPattern)
- virtual int [EncodeSingleStepTransferRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, SingleStepTransferInfo sstInfo)
- virtual int [EncodeSingleStepTransferRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToTransfer, string transferToDevice)
- virtual int [EncodeSnapshotCallRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToSnapshot)
- virtual int [EncodeSnapshotDeviceRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToSnapshot)
- virtual int [EncodeStartCDRTransRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, CDRTransferMode transferMode)
- virtual int [EncodeStartDataPathRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice)
- virtual int [EncodeStopCDRTransRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, CDRCrossRefID cdrCrossRefID)
- virtual int [EncodeStopDataPathRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, IOCrossRefID ioCrossRef)
- virtual int [EncodeTransferCallRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, TransferCallInfo tcInfo)
- virtual int [EncodeTransferCallRequest](#) (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID initiatedCall, ConnectionID originalCall)

Properties

- [PBXSession SessionObject](#) [get]
- [CSTAContext ThreadContext](#) [get]

3.4.1 Detailed Description

Implements CSTA phase 3 operations using BER. Note that most PBXes don't support all CSTA messages, so some methods in this class may result in an error status being returned by your PBX.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 GenericCSTAp3 (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.4.2.2 GenericCSTAp3 (PBXSession *sessionObject*)

Constructs an instance associated with the given [PBXSession](#) object.

Parameters

sessionObject A [PBXSession](#) object.

3.4.3 Member Function Documentation

3.4.3.1 virtual CSTAResponseInfo AlternateCall (ConnectionID *heldCall*, ConnectionID *activeCall*) [virtual]

Places an existing active call on hold and then either retrieves a previously held call or connects to an alerting or queued call at the same device (e.g., to answer a call waiting call).

Parameters

heldCall The held call to be retrieved.

activeCall The active call to be placed on hold.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.2 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.

3.4.3.3 virtual CSTAResponseInfo AnswerCall (ConnectionID *callToAnswer*) [virtual]

Answers a call.

Parameters

callToAnswer The ConnectionID of the call to answer.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.4 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.

3.4.3.5 virtual CSTAResponseInfo ClearConnection (ConnectionID *connToClear*) [virtual]

Clears a connection.

Parameters

connToClear The ConnectionID of the connection to clear.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.6 virtual CSTAResponseInfo ClearDoNotDisturb (string *targetDevice*) [virtual]

Turns off the Do Not Disturb functionality for a phone.

Parameters

targetDevice The device for which the Do Not Disturb functionality is to be turned off.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.7 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 [CSTARResponseInfo](#) object.

3.4.3.8 virtual CSTARResponseInfo ConferenceCall (ConnectionID *heldCall*, ConnectionID *activeCall*) [virtual]

Brings a held call into conference with an active call.

Parameters

heldCall The held call to be brought into conference.

activeCall The active call.

Returns

A [CSTARResponseInfo](#) object.

3.4.3.9 virtual CSTARResponseInfo 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 [CSTARResponseInfo](#) object.

3.4.3.10 virtual int EncodeACSEConnectionRequest (CSTARResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*) [protected, virtual]

Encodes an ACSE Association Request message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented in [PanasonicKXTDE](#), [SiemensCap](#), [SiemensHipath3000p3](#), and [UnifyOpenscape4000BER](#).

3.4.3.11 virtual int EncodeACSEReleaseRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*) [protected, virtual]

Encodes an ACSE Release Request message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.12 virtual int EncodeAlternateCallRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, ConnectionID *heldCall*, ConnectionID *activeCall*) [protected, virtual]

Encodes an AlternateCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

heldCall ConnectionID of the held call to be retrieved.

activeCall ConnectionID of the active call to be placed on hold.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.13 virtual int EncodeAnswerCallRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, ConnectionID *callToAnswer*, string *deviceToLift*) [protected, virtual]

Encodes an AnswerCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

callToAnswer The ConnectionID of the call to answer.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.14 `virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer) [protected, virtual]`

Encodes an AnswerCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

callToAnswer The ConnectionID of the call to answer.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.15 `virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToLift) [protected, virtual]`

Encodes an AnswerCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.16 `virtual int EncodeClearConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID connToClear) [protected, virtual]`

Encodes a ClearConnection message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

connToClear The ConnectionID of the connection to clear.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.17 `virtual int EncodeConferenceCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID heldCall, ConnectionID activeCall) [protected, virtual]`

Encodes a ConferenceCall message.

Parameters

response A CSTA ResponseInfo object.
encodeBuffer An encode buffer object into which the message will be encoded.
heldCall ConnectionID of the held call to be retrieved.
activeCall ConnectionID of the active call.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.18 **virtual int EncodeConsultationCallRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, ConnectionID *existingCall*, string *targetDevice*) [protected, virtual]**

Encodes a ConsultationCall message.

Parameters

response A CSTA ResponseInfo object.
encodeBuffer An encode buffer object into which the message will be encoded.
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

The length of the encoded message, or -1 if an error occurred.

3.4.3.19 **virtual int EncodeGetAgentStateRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, string *agentDevice*) [protected, virtual]**

Encodes a GetAgentState message.

Parameters

response A CSTA ResponseInfo object.
encodeBuffer An encode buffer object into which the message will be encoded.
agentDevice The device associated with the agent.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.20 **virtual int EncodeGetDNDRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, string *targetDevice*) [protected, virtual]**

Encodes a GetDoNotDisturb message.

Parameters

response A CSTA ResponseInfo object.
encodeBuffer An encode buffer object into which the message will be encoded.

targetDevice The phone for which the Do Not Disturb setting is desired.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.21 `virtual int EncodeGetLogicalDevInfoRequest (CSTARResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice) [protected, virtual]`

Encodes a GetLogicalDeviceInformation message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

targetDevice The device for which the information is needed.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.22 `virtual int EncodeGetPhysicalDevInfoRequest (CSTARResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice) [protected, virtual]`

Encodes a GetPhysicalDeviceInformation message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

targetDevice The device for which the information is needed.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.23 `virtual int EncodeGetSFDevicesRequest (CSTARResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ReqDeviceCategory deviceCategory) [protected, virtual]`

Encodes a GetSwitchingFunctionDevices message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.24 virtual int EncodeHoldCallRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, ConnectionID *callToHold*) [protected, virtual]

Encodes a HoldCall message.

Parameters

response A [CSTAResponseInfo](#) object.

encodeBuffer An encode buffer object into which the message will be encoded.

callToHold The ConnectionID of the call to be held.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.25 virtual int EncodeMakeCallRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, string *callingDevice*, string *calledDevice*) [protected, virtual]

Encodes a MakeCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

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

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented in [PanasonicKXTDE](#).

3.4.3.26 virtual int EncodeMonitorStartRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, ConnectionID *callToMonitor*) [protected, virtual]

Encodes a MonitorStart message to monitor a call.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

callToMonitor The call to monitor.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.27 virtual int EncodeMonitorStartRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, long *deviceToMonitor*) [protected, virtual]

Encodes a MonitorStart message to monitor a device.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

deviceToMonitor Device number of the device to monitor.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.28 virtual int EncodeMonitorStartRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, string *deviceToMonitor*) [protected, virtual]

Encodes a MonitorStart message to monitor a device.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.29 virtual int EncodeMonitorStopRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, MonitorCrossRefID *crossRefObj*) [protected, virtual]

Encodes a MonitorStop message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.30 virtual int EncodeMonitorStopRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, int *crossRefInt*, out MonitorCrossRefID *crossRefID*) [protected, virtual]

Encodes a MonitorStop message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

crossRefID A reference to a variable of type MonitorCrossRefID that will be populated with a MonitorCrossRefID object that corresponds to the specified integer cross reference id.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.31 virtual int EncodeRequestSysStatRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer) [protected, virtual]

Encodes a RequestSystemStatus message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.32 virtual int EncodeRetrieveCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToRetrieve) [protected, virtual]

Encodes a RetrieveCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

callToRetrieve The ConnectionID of the call to retrieve.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.33 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.

3.4.3.34 **virtual int EncodeSendDataRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, IOCrossRefID *ioCrossRef*, string *text*) [protected, virtual]**

Encodes a SendData message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

text The text to send to the telephony device.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.35 **virtual int EncodeSendStoredCDRRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, CDRCrossRefID *cdrCrossRefID*) [protected, virtual]**

Encodes a SendStoredCallDetailRecords message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

cdrCrossRefID The CDR cross reference id that was returned in the response to a previously issued [StartCDR-Transmission\(\)](#) call.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.36 **virtual int EncodeSetAgentStateRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, string *agentDevice*, ReqAgentState *agentState*) [protected, virtual]**

Encodes a SetAgentState message.

Parameters

response A [CSTAResponseInfo](#) object.

encodeBuffer An encode buffer object into which the message will be encoded.

agentDevice The device associated with the agent.

agentState An ReqAgentState object indicating the desired state of the agent.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.37 `virtual int EncodeSetAgentStateRequest (CSTARResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string agentDevice, ReqAgentState agentState, AgentID aid) [protected, virtual]`

Encodes a SetAgentState message.

Parameters

response A [CSTARResponseInfo](#) object.
encodeBuffer An encode buffer object into which the message will be encoded.
agentDevice The device associated with the agent.
agentState An ReqAgentState object indicating the desired state of the agent.
aid The agent id as an AgentID object.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.38 `virtual int EncodeSetDisplayRequest (CSTARResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, string text) [protected, virtual]`

Encodes a SetDisplay message.

Parameters

response A [CSTA ResponseInfo](#) object.
encodeBuffer An encode buffer object into which the message will be encoded.
targetDevice The device to which the text is to be sent.
text The text to be sent.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented in [PanasonicKXTDE](#).

3.4.3.39 `virtual int EncodeSetOrClearDNDRequest (CSTARResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool dndOn) [protected, virtual]`

Encodes a SetDoNotDisturb message.

Parameters

response A [CSTA ResponseInfo](#) object.
encodeBuffer An encode buffer object into which the message will be encoded.
targetDevice The device for which Do Not Disturb is to be set or cleared.
dndOn If true, DoNotDisturb will be turned on. If false, DoNotDisturb will be turned off.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.40 virtual int EncodeSetOrClearMWRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, string *targetDevice*, bool *indicatorOn*) [protected, virtual]

Encodes a SetMessageWaiting message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

indicatorOn If true, the message waiting indicator is turned on. If false, the message waiting indicator is turned off.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.41 virtual int EncodeSetRingerStatusRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, string *targetDevice*, string *targetRinger*, RingMode *mode*, long *ringPattern*) [protected, virtual]

Encodes a SetRingerStatus message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

targetDevice The device whose ringer is to be affected.

targetRinger The id of the ringer to use for the operation. 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, "'00000000100001010000000101'B").

mode A RingMode object that specifies ringing or notRinging.

ringPattern The indicator of the ring pattern to use.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.42 virtual int EncodeSingleStepTransferRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, SingleStepTransferInfo *sstInfo*) [protected, virtual]

Encodes a SingleStepTransfer message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

sstInfo A [SingleStepTransferInfo](#) object.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented in [SiemensHipath3000p3](#).

3.4.3.43 `virtual int EncodeSingleStepTransferRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToTransfer, string transferToDevice) [protected, virtual]`

Encodes a SingleStepTransfer message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

callToTransfer The ConnectionID of the call to transfer.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.44 `virtual int EncodeSnapshotCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToSnapshot) [protected, virtual]`

Encodes a SnapshotCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

callToSnapshot The ConnectionID of the call for which the snapshot is desired.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.45 `virtual int EncodeSnapshotDeviceRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToSnapshot) [protected, virtual]`

Encodes a SnapshotDevice message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.46 `virtual int EncodeStartCDRTransRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, CDRTransferMode transferMode) [protected, virtual]`

Encodes a StartCallDetailRecordsTransmission message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

transferMode The mode the PBX is to use to transfer call detail records.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.47 `virtual int EncodeStartDataPathRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice) [protected, virtual]`

Encodes a StartDataPath message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.48 `virtual int EncodeStopCDRTransRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, CDRCrossRefID cdrCrossRefID) [protected, virtual]`

Encodes a StopCallDetailRecordsTransmission message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

cdrCrossRefID The CDR cross reference id that was returned in the response to a previously issued [StartCDR-Transmission\(\)](#) call.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.49 `virtual int EncodeStopDataPathRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, IOCrossRefID ioCrossRef) [protected, virtual]`

Encodes a StopDataPath message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.50 virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, TransferCallInfo tcInfo) [protected, virtual]

Encodes a TransferCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

tcInfo A [TransferCallInfo](#) object.

Returns

The length of the encoded message, or -1 if an error occurred.

3.4.3.51 virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID initiatedCall, ConnectionID originalCall) [protected, virtual]

Encodes a TransferCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

The length of the encoded message, or -1 if an error occurred.

3.4.3.52 virtual CSTAResponseInfo GetAgentState (string agentDevice) [virtual]

Gets the state of the agent associated with a device.

Parameters

agentDevice The device associated with the agent.

Returns

A [CSTARResponseInfo](#) object.

3.4.3.53 virtual CSTARResponseInfo GetDoNotDisturb (string *targetDevice*) [virtual]

Gets the Do Not Disturb setting for a phone.

Parameters

targetDevice The phone for which the Do Not Disturb setting is desired.

Returns

A [CSTARResponseInfo](#) object.

3.4.3.54 virtual CSTARResponseInfo GetLogicalDevInfo (string *targetDevice*) [virtual]

Gets information about the logical element of a device.

Parameters

targetDevice The device for which the information is desired.

Returns

A [CSTARResponseInfo](#) object.

3.4.3.55 virtual CSTARResponseInfo GetPhysicalDevInfo (string *targetDevice*) [virtual]

Gets information about the physical element of a device.

Parameters

targetDevice The device for which the information is desired.

Returns

A [CSTARResponseInfo](#) object.

3.4.3.56 virtual CSTARResponseInfo 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 [CSTARResponseInfo](#) object.

3.4.3.57 virtual CSTAResponseInfo GetSFDevices () [virtual]

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

Returns

A [CSTAResponseInfo](#) object.

Reimplemented in [PanasonicKXTDE](#).

3.4.3.58 virtual CSTAResponseInfo HoldCall (ConnectionID *callToHold*) [virtual]

Instruct the PBX to hold a call.

Parameters

callToHold The ConnectionID of the call to be held.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.59 virtual CSTAResponseInfo MakeACSEAssociation () [virtual]

Establish an ACSE association with the PBX.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.60 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.

3.4.3.61 virtual CSTAResponseInfo MonitorStart (ConnectionID *callToMonitor*) [virtual]

Issues a MonitorStart request to the PBX to monitor a call.

Parameters

callToMonitor The call to monitor.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.62 virtual CSTAResponseInfo MonitorStart (long *deviceToMonitor*) [virtual]

Issues a MonitorStart request to the PBX to monitor a device.

Parameters

deviceToMonitor Device number of the device to monitor.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.63 virtual CSTAResponseInfo MonitorStart (string *deviceToMonitor*) [virtual]

Issues a MonitorStart request to the PBX to monitor a device.

Parameters

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

Returns

A [CSTAResponseInfo](#) object.

3.4.3.64 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.

3.4.3.65 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.

3.4.3.66 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.

3.4.3.67 virtual CSTAResponseInfo MonitorStopAtNumericDevice (long *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 device number of the monitored device.

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.

3.4.3.68 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.

3.4.3.69 virtual CSTAResponseInfo RequestSystemStatus () [virtual]

Retrieves a system status from the PBX.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.70 virtual CSTAResponseInfo RetrieveCall (ConnectionID *callToRetrieve*) [virtual]

Retrieves a held call.

Parameters

callToRetrieve The ConnectionID of the call to retrieve.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.71 **virtual CSTAResponseInfo 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, "0000000010000101000000101'B").

ringPattern The indicator of the ring pattern to use.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.72 **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.

3.4.3.73 **virtual CSTAResponseInfo SendStoredCDR (CDRCrossRefID *cdrCrossRefID*) [virtual]**

Issues a SendStoredCallDetailRecords request to the PBX. A CDRCallback method (see [PBXSession.CDRCallback](#)) must be defined in order to receive CDR messages.

Parameters

cdrCrossRefID The CDR cross reference id that was returned in the response to a previously issued [StartCDR-Transmission\(\)](#) call.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.74 **virtual CSTAResponseInfo SetAgentState (string *agentDevice*, ReqAgentState *agentState*) [virtual]**

Sets the state of the agent associated with a device.

Parameters

agentDevice The device associated with the agent.

agentState The desired state for the agent.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.75 virtual CSTAResponseInfo SetAgentState (string *agentDevice*, ReqAgentState *agentState*, AgentID *aid*) [virtual]

Sets the state of the agent associated with a device.

Parameters

agentDevice The device associated with the agent.

agentState The desired state for the agent.

aid The agent id.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.76 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.

3.4.3.77 virtual CSTAResponseInfo SetDoNotDisturb (string *targetDevice*) [virtual]

Sets the Do Not Disturb feature for a phone.

Parameters

targetDevice The device for which Do Not Disturb is to be set.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.78 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.

3.4.3.79 virtual CSTAResponseInfo SingleStepTransfer (SingleStepTransferInfo *sstInfo*) [virtual]

Perform a single step transfer.

Parameters

sstInfo A [SingleStepTransferInfo](#) object.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.80 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.

3.4.3.81 virtual CSTAResponseInfo SnapshotCall (ConnectionID *callToSnapshot*) [virtual]

Instruct the PBX to take a snapshot of a call.

Parameters

callToSnapshot The ConnectionID of the call for which the snapshot is desired.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.82 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.

3.4.3.83 **virtual CSTAResponseInfo StartCDRTransmission (CDRTransferMode *transferMode*)** **[virtual]**

Issues a StartCallDetailRecordsTransmission request to the PBX. A CDR callback method (see [PBXSession.CDRCallback](#)) must be defined in order to receive CDR messages.

Parameters

transferMode Indicates how the PBX is to transfer the CDR information.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.84 **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.

3.4.3.85 **virtual CSTAResponseInfo StopCDRTransmission (CDRCrossRefID *cdrCrossRefID*)** **[virtual]**

Issues a StopCallDetailRecordsTransmission request to the PBX.

Parameters

cdrCrossRefID The CDR cross reference id that was returned in the response to a previously issued [StartCDRTransmission\(\)](#) call.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.86 **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.

3.4.3.87 **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.

3.4.3.88 **virtual CSTAResponseInfo TransferCall (TransferCallInfo *tcInfo*)** **[virtual]**

Transfers a call from one device to another.

Parameters

tcInfo A [TransferCallInfo](#) object.

Returns

A [CSTAResponseInfo](#) object.

3.4.3.89 **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.

3.4.4 **Property Documentation**

3.4.4.1 **PBXSession SessionObject** **[get]**

The [PBXSession](#) object associated with this instance.

3.4.4.2 **CSTAContext ThreadContext** **[get]**

The [CSTAContext](#) structure for this thread.

3.5 IETF_CSTAp3 Class Reference

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

Inherited by [PanasonicKXTDE](#), [SiemensCap](#), and [UnifyOpenscape4000BER](#).

Public Member Functions

- [IETF_CSTAp3](#) ([PBXSession](#) sessionObject)
- [IETF_CSTAp3](#) (string pbxSystem, int port)

3.5.1 Detailed Description

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

3.5.2 Constructor & Destructor Documentation

3.5.2.1 IETF_CSTAp3 (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.5.2.2 IETF_CSTAp3 ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given [PBXSession](#) object.

Parameters

sessionObject A [PBXSession](#) object.

3.6 LicenseException Class Reference

3.6.1 Detailed Description

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

3.7 LicenseOptions Class Reference

3.7.1 Detailed Description

This class holds booleans that define what capabilities are defined in the license.

3.8 PanasonicKXTDA Class Reference

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

Public Member Functions

- [PanasonicKXTDA](#) ([PBXSession](#) sessionObject)
- [PanasonicKXTDA](#) (string pbxSystem, int port)

3.8.1 Detailed Description

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

3.8.2 Constructor & Destructor Documentation

3.8.2.1 [PanasonicKXTDA](#) (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

- pbxSystem* Well-known name or IP address of the PBX.
port Port on which the PBX listens for CSTA messages.

3.8.2.2 [PanasonicKXTDA](#) ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given PBXSession object.

Parameters

- sessionObject* A PBXSession object.

3.9 PanasonicKXTDE Class Reference

Inherits [Com::Objsys::Csta::Phase3::IETF_CSTAp3](#).

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

Public Types

- enum [DeviceDataTypes](#)

Public Member Functions

- [CSTAResponseInfo AcquireControlRight](#) (string targetDevice)
- [CSTAResponseInfo ClearMessageWaiting](#) (string originatingDevice, string targetDevice)
- [CSTAResponseInfo GetDeviceData](#) (string device, [DeviceDataTypes](#) eDataType)
- [CSTAResponseInfo GetGroupMembers](#) (string groupDevice)
- override [CSTAResponseInfo GetSFDevices](#) ()
- [PanasonicKXTDE](#) ([PBXSession](#) sessionObject)
- [PanasonicKXTDE](#) (string pbxSystem, int port)
- [CSTAResponseInfo PDFStart](#) (string targetDevice)
- [CSTAResponseInfo PDFStop](#) (string targetDevice)
- [CSTAResponseInfo ReleaseControlRight](#) (string targetDevice)
- [CSTAResponseInfo ResetDisplay](#) (string targetDevice)
- [CSTAResponseInfo SendKmeMessage](#) ([Asn1BerEncodeBuffer](#) encodeBuffer)
- [CSTAResponseInfo SetMessageWaiting](#) (string originatingDevice, string targetDevice)

Protected Member Functions

- override int [EncodeACSEConnectionRequest](#) ([CSTAResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer)
- override int [EncodeMakeCallRequest](#) ([CSTAResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer, string callingDevice, string calledDevice)
- override int [EncodeSetDisplayRequest](#) ([CSTAResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer, string targetDevice, string text)

3.9.1 Detailed Description

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

3.9.2 Member Enumeration Documentation

3.9.2.1 enum DeviceDataTypes

Provides symbolic names for the types of device information that can be obtained from a Panasonic PBX.

3.9.3 Constructor & Destructor Documentation

3.9.3.1 PanasonicKXTDE (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.9.3.2 PanasonicKXTDE (PBXSession *sessionObject*)

Constructs an instance associated with the given PBXSession object.

Parameters

sessionObject A PBXSession object.

3.9.4 Member Function Documentation

3.9.4.1 CSTAResponseInfo 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 CSTAResponseInfo object.

3.9.4.2 CSTAResponseInfo 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 CSTAResponseInfo object.

3.9.4.3 override int EncodeACSEConnectionRequest (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*) [protected, virtual]

Encodes an ACSE Association Request message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented from [GenericCSTAp3](#).

3.9.4.4 **override int EncodeMakeCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice) [protected, virtual]**

Encodes a MakeCall message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

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

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented from [GenericCSTAp3](#).

3.9.4.5 **override int EncodeSetDisplayRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, string text) [protected, virtual]**

Encodes a SetDisplay message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

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

text The text to be sent.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented from [GenericCSTAp3](#).

3.9.4.6 **CSTAResponseInfo GetDeviceData (string device, DeviceDataTypes eDataType)**

Gets information about a device.

Parameters

device The device (e.g., "101") about which the information is desired.

eDataType The type of data requested (must be from the DeviceDataTypes enum in this class).

Returns

3.9.4.7 CSTAResponseInfo 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 CSTAResponseInfo object.

3.9.4.8 override CSTAResponseInfo 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 CSTAResponseInfo object.

Reimplemented from [GenericCSTAp3](#).

3.9.4.9 CSTAResponseInfo 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 CSTAResponseInfo object.

3.9.4.10 CSTAResponseInfo 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 CSTAResponseInfo object.

3.9.4.11 CSTAResponseInfo 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 CSTAResponseInfo object.

3.9.4.12 CSTAResponseInfo ResetDisplay (string *targetDevice*)

Resets a telephony device's display

Parameters

targetDevice The device which is to be reset.

Returns

A CSTAResponseInfo object.

3.9.4.13 CSTAResponseInfo 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

3.9.4.14 CSTAResponseInfo 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 CSTAResponseInfo object.

3.10 PanasonicNCP Class Reference

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

Public Member Functions

- [PanasonicNCP](#) ([PBXSession](#) sessionObject)
- [PanasonicNCP](#) (string pbxSystem, int port)

3.10.1 Detailed Description

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

3.10.2 Constructor & Destructor Documentation

3.10.2.1 [PanasonicNCP](#) (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.10.2.2 [PanasonicNCP](#) ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given PBXSession object.

Parameters

sessionObject A PBXSession object.

3.11 PBXSession Class Reference

Public Member Functions

- delegate void [AsyncCallback](#) ([PBXSession](#) sessionObject, byte[] asyncData)
- delegate void [AsyncExceptionCallback](#) ([PBXSession](#) sessionObject, ApplicationException exception)
- void [Close](#) ([CSTAContext](#) threadContext)
- delegate void [ConnectionCallback](#) ([PBXSession](#) sessionObject)
- void [Open](#) ([CSTAContext](#) threadContext)
- [PBXSession](#) (string pbxSystem, int port)
- [SocketState](#) [SendACSEMessage](#) (byte[] message, int messageLength, [Constants.ACSEMessageTypes](#) messageType, [CSTAContext](#) threadContext)
- void [SendMessage](#) (string messageType, byte[] message, int messageLength, [CSTAContext](#) threadContext)
- void [SendMessage](#) (byte[] message, int messageLength, [CSTAContext](#) threadContext)
- void [SendXMLMessage](#) (string messageType, string strMessage, [CSTAContext](#) threadContext)
- void [SendXMLMessage](#) (string strMessage, [CSTAContext](#) threadContext)
- [SocketState](#) [SendXMLSession](#) (string strMessage, [Constants.XMLSessionMessageTypes](#) enmMessageType, [CSTAContext](#) threadContext)
- void [WaitForROSEResponse](#) ([CSTAContext](#) threadContext)
- void [WaitForXMLResponse](#) ([CSTAContext](#) threadContext)
- delegate void [XMLAsyncCallback](#) ([PBXSession](#) sessionObject, string message)

Properties

- [Constants.CallbackInvocationMechanisms](#) [CallbackInvocationMechanism](#) [get, set]
- AsyncCallback [CDRCallback](#) [get, set]
- AsyncCallback [ClientCallback](#) [get, set]
- bool [Connected](#) [get, set]
- ConnectionCallback [ConnectionLostCallback](#) [get, set]
- bool [DebugMode](#) [get, set]
- AsyncExceptionCallback [ExceptionCallback](#) [get, set]
- int [MaxReceiveTimeout](#) [get, set]
- [Constants.Encoding](#) [MessageEncoding](#) [get, set]
- string [PBXSystem](#) [get]
- int [Port](#) [get]
- AsyncCallback [SystemStatusCallback](#) [get, set]
- XMLAsyncCallback [XMLCDRCallback](#) [get, set]
- XMLAsyncCallback [XMLClientCallback](#) [get, set]
- XMLAsyncCallback [XMLSystemStatusCallback](#) [get, set]

3.11.1 Detailed Description

This class manages communication with a PBX. One instance of this class should be created for each PBX with which a CSTADLL client application needs to exchange CSTA messages.

The CSTA worker classes (e.g., Alcatel4400, PanasonicNCP) hold a reference to a [PBXSession](#) object. If the constructor for the worker class that takes a PBX identification and a PBX port is used, a [PBXSession](#) object is created. Alternatively, the client application can create a [PBXSession](#) instance and pass a reference to the instance to the other worker class constructor signature.

Only one [PBXSession](#) instance for a PBX/port combination should be created. The behavior is undefined if multiple [PBXSession](#) instances are created for the same PBX and port.

3.11.2 Constructor & Destructor Documentation

3.11.2.1 PBXSession (string *pbxSystem*, int *port*)

Constructs a [PBXSession](#) object.

Parameters

- pbxSystem* The name or IP address of the PBX system.
port The port on the PBX system to which the client is connecting.

3.11.3 Member Function Documentation

3.11.3.1 delegate void AsyncCallback (PBXSession *sessionObject*, byte[] *asyncData*)

Declaration of a callback function to be invoked when one of the following messages is received: a monitor event report message, a route message, a CDR Report message, a CDR Notification message, or a system status request.

Parameters

- sessionObject* The session object for the PBX that generated the asynchronous message.
asyncData The data received asynchronously from the PBX.

3.11.3.2 delegate void AsyncExceptionCallback (PBXSession *sessionObject*, ApplicationException *exception*)

Declaration of a callback function to be invoked if a condition is encountered in the asynchronous I/O handler that would otherwise result in an exception being thrown. Note that in a couple of cases the asynchronous code will still throw an exception, even if this callback is defined.

Parameters

- sessionObject* The session object for the PBX that sent a packet that triggered an exception condition.
exception The ApplicationException object that would have been thrown in the asynchronous I/O handling code if this callback were not defined.

3.11.3.3 void Close (CSTAContext *threadContext*)

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.

3.11.3.4 delegate void ConnectionCallback (PBXSession *sessionObject*)

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

Parameters

- sessionObject* The session object for the PBX whose connection was lost.

3.11.3.5 void Open (CSTAContext *threadContext*)

This method can be used to establish communication with a PBX device before any messages are actually sent to the device. TCP/IP connectivity is established and an asynchronous read is started to receive messages sent from the PBX.

Parameters

threadContext The thread context object.

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

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.

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

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.

3.11.3.8 void SendMessage (byte[] *message*, int *messageLength*, CSTAContext *threadContext*)

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.

3.11.3.9 void SendXMLMessage (string *messageType*, string *strMessage*, CSTAContext *threadContext*)

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

Parameters

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

strMessage The XML message to send.

threadContext The thread context object.

3.11.3.10 void SendXMLMessage (string *strMessage*, CSTAContext *threadContext*)

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

Parameters

strMessage The XML message to send.

threadContext The thread context object.

3.11.3.11 SocketState SendXMLSession (string *strMessage*, Constants.XMLSessionMessageTypes *enmMessageType*, CSTAContext *threadContext*)

This method sends an XML session management (ECMA-354) message to the PBX.

Parameters

strMessage The text of the XML message to send.

enmMessageType A constant indicating what kind of session management message is being sent.

threadContext The thread context object.

Returns

A populated [SocketState](#) instance if the message is a StartSession message. Null if the message is StopSession or ResetSession.

3.11.3.12 void WaitForROSEResponse (CSTAContext *threadContext*)

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.

3.11.3.13 void WaitForXMLResponse (CSTAContext *threadContext*)

This method waits for a response to an XML CSTA message.

Parameters

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

3.11.3.14 **delegate void XMLAsyncCallback (PBXSession *sessionObject*, string *message*)**

Declaration of a callback function to be invoked when an asynchronous XML monitor event or route message is received.

Parameters

sessionObject The session object for the PBX or UA that generated the asynchronous message.

message The text of the message received asynchronously from the PBX or UA.

3.11.4 **Property Documentation**

3.11.4.1 **Constants.CallbackInvocationMechanisms CallbackInvocationMechanism [get, set]**

Indicates what asynchronous callback invocation mechanism to use. See the documentation on [Constants.CallbackInvocationMechanisms](#) for a detailed description of the possible options.

3.11.4.2 **AsyncCallback CDRCallback [get, set]**

Holds a reference to an asynchronous callback function that will be invoked when CDR information is received asynchronously. This information could take the form of a CDR Report message or a CDR Notification message. If the latter, the application should use the SendStoredCDR() method to request the stored CDRs from the PBX.

3.11.4.3 **AsyncCallback ClientCallback [get, set]**

Holds a reference to an asynchronous callback function. This function will be invoked if a monitor event or route message is received asynchronously from the PBX.

3.11.4.4 **bool Connected [get, set]**

Indicates whether the session to the PBX is connected.

3.11.4.5 **ConnectionCallback ConnectionLostCallback [get, set]**

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

3.11.4.6 **bool DebugMode [get, set]**

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

3.11.4.7 **AsyncExceptionCallback ExceptionCallback [get, set]**

Holds a reference to an asynchronous callback function. This function will be invoked if a condition occurs in the asynchronous I/O handler that otherwise would have resulted in an exception being thrown if this callback were not defined. Note that in a couple of cases the asynchronous code will still throw an exception, even if this callback is defined.

3.11.4.8 int MaxReceiveTimeout [get, set]

Specifies the amount of time, in milliseconds, to wait for a response to arrive from the PBX. The default value is 5,000 milliseconds (5 seconds).

3.11.4.9 Constants.Encoding MessageEncoding [get, set]

Indicates how messages exchanges with this PBX are encoded.

3.11.4.10 string PBXSystem [get]

The TCIP/IP address or well-known name of the PBX.

3.11.4.11 int Port [get]

The port where the PBX listens for CSTA messages.

3.11.4.12 AsyncCallback SystemStatusCallback [get, set]

Holds a reference to an asynchronous callback function. This function will be invoked if a BER System Status message is received asynchronously from the PBX. The client application does NOT need to send a System Status Response, since that is taken care of by CSTADLL.

3.11.4.13 XMLAsyncCallback XMLCDRCallback [get, set]

Holds a reference to an asynchronous callback function that will be invoked when XML CDR information is received asynchronously.

This information could take the form of a CDR Report message or a CDR Notification message. If the latter, the application should use the SendStoredCDR() method to request the stored CDRs from the PBX or UA.

3.11.4.14 XMLAsyncCallback XMLClientCallback [get, set]

Holds a reference to an asynchronous XML callback function. This function will be invoked if an XML monitor event report message or route message is received asynchronously from the PBX or UA.

3.11.4.15 XMLAsyncCallback XMLSystemStatusCallback [get, set]

Holds a reference to an asynchronous callback function. This function will be invoked if an XML System Status message is received asynchronously from the PBX. The client application does NOT need to send a System Status Response, since that is taken care of by CSTADLL.

3.12 PBXSessionException Class Reference

3.12.1 Detailed Description

Defines an exception that occurs while communicating with a PBX.

3.13 PBXSessionHelper Class Reference

Properties

- static bool [LoggingEnabled](#) [get, set]
- static string [LoggingFolder](#) [get, set]
- static long [MaxLogFileSize](#) [get, set]

3.13.1 Detailed Description

This class holds static properties that affect all PBX sessions.

3.13.2 Property Documentation

3.13.2.1 bool `LoggingEnabled` [static, get, set]

Indicates whether logging should be done.

3.13.2.2 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.

3.13.2.3 long `MaxLogFileSize` [static, get, set]

Defines the maximum size, in bytes, that a log file is allowed to grow to before a new log file is opened. If no value is specified for this property, the maximum size is Constants.MAX_LOG_FILE_SIZE. Any value specified for this property overrides this default setting.

3.14 PBXSessionHelperPhase3 Class Reference

3.14.1 Detailed Description

This class contains utility methods used by [PBXSessionHelper](#) for BER phase 3.

3.15 Phase3Opcodes Class Reference

Public Types

- enum [Opcodes](#)

3.15.1 Detailed Description

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

3.15.2 Member Enumeration Documentation

3.15.2.1 enum Opcodes

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

3.16 ROSEParseInfo Class Reference

3.16.1 Detailed Description

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

3.17 SiemensCap Class Reference

Inherits [Com::Objsys::Csta::Phase3::IETF_CSTAp3](#).

Inherited by [SiemensHipath4000](#).

Public Member Functions

- [CSTAResponseInfo MakeACSEAssociation](#) (string appid, string login, string passwd, bool nativeMode, CSTAVersion cv)
- [CSTAResponseInfo MakeACSEAssociation](#) (string appid, string login, string passwd, bool nativeMode)
- [CSTAResponseInfo MakeACSEAssociation](#) (string login, string passwd)
- [SiemensCap](#) ([PBXSession](#) sessionObject)
- [SiemensCap](#) (string pbxSystem, int port)

Protected Member Functions

- override int [EncodeACSEConnectionRequest](#) ([CSTAResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer)

3.17.1 Detailed Description

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

3.17.2 Constructor & Destructor Documentation

3.17.2.1 [SiemensCap](#) (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.17.2.2 [SiemensCap](#) ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given PBXSession object.

Parameters

sessionObject A PBXSession object.

3.17.3 Member Function Documentation

3.17.3.1 override int [EncodeACSEConnectionRequest](#) ([CSTAResponseInfo](#) *response*, [Asn1BerEncodeBuffer](#) *encodeBuffer*) [[protected](#), [virtual](#)]

Encodes an ACSE Association Request message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented from [GenericCSTAp3](#).

3.17.3.2 CSTAResponseInfo MakeACSEAssociation (string *appid*, string *login*, string *passwd*, bool *nativeMode*, CSTAVersion *cv*)

Establish an ACSE association with the PBX using the specified arguments.

Parameters

appid The application id to be encoded into the ACSE Association Request.

login The login to be encoded into the ACSE Association Request.

passwd The password to be encoded into the ACSE Association Request.

nativeMode If true, "native=true" will be included in the ACSE Association Request. If false, "native=true" will not be included.

cv A CSTAVersion object representing the CSTA version to encode into the ACSE Association Request.

Returns

A CSTAResponseInfo object.

3.17.3.3 CSTAResponseInfo MakeACSEAssociation (string *appid*, string *login*, string *passwd*, bool *nativeMode*)

Establish an ACSE association with the PBX using the specified arguments.

Parameters

appid The application id to be encoded into the ACSE Association Request.

login The login to be encoded into the ACSE Association Request.

passwd The password to be encoded into the ACSE Association Request.

nativeMode If true, "native=true" will be included in the ACSE Association Request. If false, "native=true" will not be included.

Returns

A CSTAResponseInfo object.

3.17.3.4 CSTAResponseInfo MakeACSEAssociation (string *login*, string *passwd*)

Establish an ACSE association with the PBX using a specified login and password.

Parameters

login The login to be encoded into the ACSE Association Request.

passwd The password to be encoded into the ACSE Association Request.

Returns

A CSTAResponseInfo object.

3.18 SiemensHipath3000p3 Class Reference

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

Inherited by [UnifyOpenscapeX5](#).

Public Member Functions

- virtual [CSTAResponseInfo MakeACSEAssociation](#) (string userName, string password)
- [SiemensHipath3000p3](#) ([PBXSession](#) sessionObject)
- [SiemensHipath3000p3](#) (string pbxSystem, int port)

Protected Member Functions

- override int [EncodeACSEConnectionRequest](#) ([CSTAResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer)
- override int [EncodeSingleStepTransferRequest](#) ([CSTAResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer, [SingleStepTransferInfo](#) sstInfo)

3.18.1 Detailed Description

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

3.18.2 Constructor & Destructor Documentation

3.18.2.1 [SiemensHipath3000p3](#) (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

- pbxSystem* Well-known name or IP address of the PBX.
- port* Port on which the PBX listens for CSTA messages.

3.18.2.2 [SiemensHipath3000p3](#) ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given [PBXSession](#) object.

Parameters

- sessionObject* A [PBXSession](#) object.

3.18.3 Member Function Documentation

3.18.3.1 override int [EncodeACSEConnectionRequest](#) ([CSTAResponseInfo](#) *response*, [Asn1BerEncodeBuffer](#) *encodeBuffer*) [[protected](#), [virtual](#)]

Encodes an ACSE Association Request message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented from [GenericCSTAp3](#).

3.18.3.2 `override int EncodeSingleStepTransferRequest (CSTAResponseInfo response,
Asn1BerEncodeBuffer encodeBuffer, SingleStepTransferInfo sstInfo) [protected, virtual]`

Encodes a SingleStepTransfer message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

sstInfo A SingleStepTransferInfo object.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented from [GenericCSTAp3](#).

3.18.3.3 `virtual CSTAResponseInfo MakeACSEAssociation (string userName, string password)
[virtual]`

Establish an ACSE association with the PBX.

Parameters

userName The user name to send to the PBX.

password The password to send to the PBX.

Returns

A CSTAResponseInfo object.

3.19 SiemensHipath4000 Class Reference

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

Public Member Functions

- [SiemensHipath4000](#) ([PBXSession](#) sessionObject)
- [SiemensHipath4000](#) (string pbxSystem, int port)

3.19.1 Detailed Description

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

3.19.2 Constructor & Destructor Documentation

3.19.2.1 SiemensHipath4000 (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.19.2.2 SiemensHipath4000 ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given [PBXSession](#) object.

Parameters

sessionObject A [PBXSession](#) object.

3.20 SingleStepTransferInfo Class Reference

Properties

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

3.20.1 Detailed Description

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

3.20.2 Property Documentation

3.20.2.1 string TransferFromDevice [get, set]

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

3.20.2.2 string TransferringCallID [get, set]

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

3.20.2.3 string TransferToDevice [get, set]

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

3.21 SocketState Class Reference

Properties

- `byte[] AckBuffer` [get, set]
- `byte[] ReadBuffer` [get, set]
- `List< byte[] > ReadBuffers` [get, set]
- `int TotalLength` [get, set]

3.21.1 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.

3.21.2 Property Documentation

3.21.2.1 `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.

3.21.2.2 `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.

3.21.2.3 `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.

3.21.2.4 `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.

3.22 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]

3.22.1 Detailed Description

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

3.22.2 Property Documentation

3.22.2.1 string ActiveDeviceCallID [get, set]

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

3.22.2.2 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.

3.22.2.3 string ConnectedDevice [get, set]

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

3.22.2.4 string HeldDevice [get, set]

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

3.22.2.5 string HeldDeviceCallID [get, set]

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

3.22.2.6 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.

3.23 UnifyOpenscape4000BER Class Reference

Inherits [Com::Objsys::Csta::Phase3::IETF_CSTAp3](#).

Public Member Functions

- virtual [CSTAResponseInfo MakeACSEAssociation](#) (string userName, string password)
- [UnifyOpenscape4000BER](#) ([PBXSession](#) sessionObject)
- [UnifyOpenscape4000BER](#) (string pbxSystem, int port)

Protected Member Functions

- override int [EncodeACSEConnectionRequest](#) ([CSTAResponseInfo](#) response, [Asn1BerEncodeBuffer](#) encodeBuffer)

3.23.1 Detailed Description

Implements CSTA phase 3 BER operations for the Unify Openscape 4000 device.

3.23.2 Constructor & Destructor Documentation

3.23.2.1 UnifyOpenscape4000BER (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.23.2.2 UnifyOpenscape4000BER ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given [PBXSession](#) object.

Parameters

sessionObject A [PBXSession](#) object.

3.23.3 Member Function Documentation

3.23.3.1 override int [EncodeACSEConnectionRequest](#) ([CSTAResponseInfo](#) *response*, [Asn1BerEncodeBuffer](#) *encodeBuffer*) [**protected**, **virtual**]

Encodes an ACSE Association Request message.

Parameters

response A CSTA ResponseInfo object.

encodeBuffer An encode buffer object into which the message will be encoded.

Returns

The length of the encoded message, or -1 if an error occurred.

Reimplemented from [GenericCSTAp3](#).

**3.23.3.2 virtual CSTAResponseInfo MakeACSEAssociation (string *userName*, string *password*)
[virtual]**

Establish an ACSE association with the PBX.

Parameters

userName The user name to send to the PBX.

password The password to send to the PBX.

Returns

A CSTAResponseInfo object.

3.24 UnifyOpenscapeX5 Class Reference

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

Public Member Functions

- [UnifyOpenscapeX5](#) ([PBXSession](#) sessionObject)
- [UnifyOpenscapeX5](#) (string pbxSystem, int port)

3.24.1 Detailed Description

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

3.24.2 Constructor & Destructor Documentation

3.24.2.1 UnifyOpenscapeX5 (string *pbxSystem*, int *port*)

Constructs an instance associated with the given PBX identifier and port.

Parameters

pbxSystem Well-known name or IP address of the PBX.

port Port on which the PBX listens for CSTA messages.

3.24.2.2 UnifyOpenscapeX5 ([PBXSession](#) *sessionObject*)

Constructs an instance associated with the given [PBXSession](#) object.

Parameters

sessionObject A [PBXSession](#) object.

Index

- AckBuffer
 - Com::Objsys::Csta::Phase3::SocketState, 66
- AcquireControlRight
 - Com::Objsys::Csta::Devices::PanasonicKXTDE, 43
- ACSEMessageTypes
 - Com::Objsys::Csta::Phase3::Constants, 5
- ActiveDeviceCallID
 - Com::Objsys::Csta::Phase3::TransferCallInfo, 67
- ActiveDeviceNumber
 - Com::Objsys::Csta::Phase3::TransferCallInfo, 67
- AlternateCall
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 13
- AnswerCall
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 13, 14
- AsyncCallback
 - Com::Objsys::Csta::Phase3::PBXSession, 49
- AsyncExceptionCallback
 - Com::Objsys::Csta::Phase3::PBXSession, 49
- CallbackInvocationMechanism
 - Com::Objsys::Csta::Phase3::PBXSession, 52
- CallbackInvocationMechanisms
 - Com::Objsys::Csta::Phase3::Constants, 5
- CDRCallback
 - Com::Objsys::Csta::Phase3::PBXSession, 52
- ClearConnection
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 14
- ClearDoNotDisturb
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 14
- ClearMessageWaiting
 - Com::Objsys::Csta::Devices::PanasonicKXTDE, 43
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 14
- ClientCallback
 - Com::Objsys::Csta::Phase3::PBXSession, 52
- Close
 - Com::Objsys::Csta::Phase3::PBXSession, 49
- Com.Objsys.Csta.Devices, 3
- Com.Objsys.Csta.Phase3, 4
- Com::Objsys::Csta::Devices::PanasonicKXTDA, 41
 - PanasonicKXTDA, 41
- Com::Objsys::Csta::Devices::PanasonicKXTDE, 42
 - AcquireControlRight, 43
 - ClearMessageWaiting, 43
 - DeviceDataTypes, 42
 - EncodeACSEConnectionRequest, 43
 - EncodeMakeCallRequest, 44
 - EncodeSetDisplayRequest, 44
 - GetDeviceData, 44
 - GetGroupMembers, 45
 - GetSFDevices, 45
 - PanasonicKXTDE, 43
 - PDFStart, 45
 - PDFStop, 45
 - ReleaseControlRight, 45
 - ResetDisplay, 46
 - SendKmeMessage, 46
 - SetMessageWaiting, 46
- Com::Objsys::Csta::Devices::PanasonicNCP, 47
 - PanasonicNCP, 47
- Com::Objsys::Csta::Devices::SiemensCap, 59
 - EncodeACSEConnectionRequest, 59
 - MakeACSEAssociation, 60
 - SiemensCap, 59
- Com::Objsys::Csta::Devices::SiemensHipath3000p3, 62
 - EncodeACSEConnectionRequest, 62
 - EncodeSingleStepTransferRequest, 63
 - MakeACSEAssociation, 63
 - SiemensHipath3000p3, 62
- Com::Objsys::Csta::Devices::SiemensHipath4000, 64
 - SiemensHipath4000, 64
- Com::Objsys::Csta::Devices::UnifyOpenscape4000BER, 68
 - EncodeACSEConnectionRequest, 68
 - MakeACSEAssociation, 69
 - UnifyOpenscape4000BER, 68
- Com::Objsys::Csta::Devices::UnifyOpenscapeX5, 70
 - UnifyOpenscapeX5, 70
- Com::Objsys::Csta::Phase3::Constants, 5
 - ACSEMessageTypes, 5
 - CallbackInvocationMechanisms, 5
 - CommunicationTypes, 6
 - Encoding, 6
 - MAX_LOGFILE_SIZE, 6
 - PBXModels, 6
 - XMLSessionMessageTypes, 6
- Com::Objsys::Csta::Phase3::CSTAContext, 7
 - ResponseFromPBX, 7
 - ResponsesFromPBX, 7
 - XMLResponseFromPBX, 7

- XMLResponsesFromPBX, 7
- Com::Objsys::Csta::Phase3::CSTAResponseInfo, 8
 - ResponseFromPBX, 8
 - ResponsesFromPBX, 8
 - ResponsesFromUA, 8
 - StatusCode, 8
 - StatusMessage, 9
 - XMLResponseFromPBX, 9
 - XMLResponseFromUA, 9
 - XMLResponsesFromPBX, 9
- Com::Objsys::Csta::Phase3::GenericCSTAp3, 10
 - AlternateCall, 13
 - AnswerCall, 13, 14
 - ClearConnection, 14
 - ClearDoNotDisturb, 14
 - ClearMessageWaiting, 14
 - ConferenceCall, 15
 - ConsultationCall, 15
 - EncodeACSEConnectionRequest, 15
 - EncodeACSEReleaseRequest, 15
 - EncodeAlternateCallRequest, 16
 - EncodeAnswerCallRequest, 16, 17
 - EncodeClearConnectionRequest, 17
 - EncodeConferenceCallRequest, 17
 - EncodeConsultationCallRequest, 18
 - EncodeGetAgentStateRequest, 18
 - EncodeGetDNDRequest, 18
 - EncodeGetLogicalDevInfoRequest, 19
 - EncodeGetPhysicalDevInfoRequest, 19
 - EncodeGetSFDevicesRequest, 19
 - EncodeHoldCallRequest, 19
 - EncodeMakeCallRequest, 20
 - EncodeMonitorStartRequest, 20, 21
 - EncodeMonitorStopRequest, 21
 - EncodeRequestSysStatRequest, 22
 - EncodeRetrieveCallRequest, 22
 - EncodeROSERequestHeader, 22
 - EncodeSendDataRequest, 23
 - EncodeSendStoredCDRRequest, 23
 - EncodeSetAgentStateRequest, 23
 - EncodeSetDisplayRequest, 24
 - EncodeSetOrClearDNDRequest, 24
 - EncodeSetOrClearMWRequest, 24
 - EncodeSetRingerStatusRequest, 25
 - EncodeSingleStepTransferRequest, 25
 - EncodeSnapshotCallRequest, 26
 - EncodeSnapshotDeviceRequest, 26
 - EncodeStartCDRTransRequest, 26
 - EncodeStartDataPathRequest, 27
 - EncodeStopCDRTransRequest, 27
 - EncodeStopDataPathRequest, 27
 - EncodeTransferCallRequest, 28
 - GenericCSTAp3, 13
 - GetAgentState, 28
 - GetDoNotDisturb, 29
 - GetLogicalDevInfo, 29
 - GetPhysicalDevInfo, 29
 - GetSFDevices, 29
 - HoldCall, 30
 - MakeACSEAssociation, 30
 - MakeCall, 30
 - MonitorStart, 30, 31
 - MonitorStop, 31
 - MonitorStopAtNumericDevice, 32
 - ReleaseACSEAssociation, 32
 - RequestSystemStatus, 32
 - RetrieveCall, 32
 - RingDevice, 32
 - SendData, 33
 - SendStoredCDR, 33
 - SessionObject, 37
 - SetAgentState, 33
 - SetDisplay, 34
 - SetDoNotDisturb, 34
 - SetMessageWaiting, 34
 - SingleStepTransfer, 34, 35
 - SnapshotCall, 35
 - SnapshotDevice, 35
 - StartCDRTransmission, 35
 - StartDataPath, 36
 - StopCDRTransmission, 36
 - StopDataPath, 36
 - StopRing, 36
 - ThreadContext, 37
 - TransferCall, 37
- Com::Objsys::Csta::Phase3::IETF_CSTAp3, 38
 - IETF_CSTAp3, 38
- Com::Objsys::Csta::Phase3::LicenseException, 39
- Com::Objsys::Csta::Phase3::LicenseOptions, 40
- Com::Objsys::Csta::Phase3::PBXSession, 48
 - AsyncCallback, 49
 - AsyncExceptionCallback, 49
 - CallbackInvocationMechanism, 52
 - CDRCallback, 52
 - ClientCallback, 52
 - Close, 49
 - Connected, 52
 - ConnectionCallback, 49
 - ConnectionLostCallback, 52
 - DebugMode, 52
 - ExceptionCallback, 52
 - MaxReceiveTimeout, 52
 - MessageEncoding, 53
 - Open, 49
 - PBXSession, 49
 - PBXSystem, 53
 - Port, 53
 - SendACSEMessage, 50

- SendMessage, 50
- SendXMLMessage, 50, 51
- SendXMLSession, 51
- SystemStatusCallback, 53
- WaitForROSEResponse, 51
- WaitForXMLResponse, 51
- XMLAsyncCallback, 51
- XMLCDRCallback, 53
- XMLClientCallback, 53
- XMLSystemStatusCallback, 53
- Com::Objsys::Csta::Phase3::PBXSessionException, 54
- Com::Objsys::Csta::Phase3::PBXSessionHelper, 55
 - LoggingEnabled, 55
 - LoggingFolder, 55
 - MaxLogFileSize, 55
- Com::Objsys::Csta::Phase3::PBXSessionHelperPhase3, 56
- Com::Objsys::Csta::Phase3::Phase3Opcodes, 57
 - Opcodes, 57
- Com::Objsys::Csta::Phase3::ROSEParseInfo, 58
- Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 65
 - TransferFromDevice, 65
 - TransferringCallID, 65
 - TransferToDevice, 65
- Com::Objsys::Csta::Phase3::SocketState, 66
 - AckBuffer, 66
 - ReadBuffer, 66
 - ReadBuffers, 66
 - TotalLength, 66
- Com::Objsys::Csta::Phase3::TransferCallInfo, 67
 - ActiveDeviceCallID, 67
 - ActiveDeviceNumber, 67
 - ConnectedDevice, 67
 - HeldDevice, 67
 - HeldDeviceCallID, 67
 - HeldDeviceNumber, 67
- CommunicationTypes
 - Com::Objsys::Csta::Phase3::Constants, 6
- ConferenceCall
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 15
- Connected
 - Com::Objsys::Csta::Phase3::PBXSession, 52
- ConnectedDevice
 - Com::Objsys::Csta::Phase3::TransferCallInfo, 67
- ConnectionCallback
 - Com::Objsys::Csta::Phase3::PBXSession, 49
- ConnectionLostCallback
 - Com::Objsys::Csta::Phase3::PBXSession, 52
- ConsultationCall
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 15
- DebugMode
 - Com::Objsys::Csta::Phase3::PBXSession, 52
- DeviceDataTypes

- Com::Objsys::Csta::Devices::PanasonicKXTDE, 42
- EncodeACSEConnectionRequest
 - Com::Objsys::Csta::Devices::PanasonicKXTDE, 43
 - Com::Objsys::Csta::Devices::SiemensCap, 59
 - Com::Objsys::Csta::Devices::SiemensHipath3000p3, 62
 - Com::Objsys::Csta::Devices::UnifyOpenscape4000BER, 68
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 15
- EncodeACSEReleaseRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 15
- EncodeAlternateCallRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 16
- EncodeAnswerCallRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 16, 17
- EncodeClearConnectionRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 17
- EncodeConferenceCallRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 17
- EncodeConsultationCallRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 18
- EncodeGetAgentStateRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 18
- EncodeGetDNDRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 18
- EncodeGetLogicalDevInfoRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 19
- EncodeGetPhysicalDevInfoRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 19
- EncodeGetSFDevicesRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 19
- EncodeHoldCallRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 19
- EncodeMakeCallRequest
 - Com::Objsys::Csta::Devices::PanasonicKXTDE, 44
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 20
- EncodeMonitorStartRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 20, 21
- EncodeMonitorStopRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 21
- EncodeRequestSysStatRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 22
- EncodeRetrieveCallRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 22
- EncodeROSERequestHeader
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 22
- EncodeSendDataRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 23
- EncodeSendStoredCDRRequest
 - Com::Objsys::Csta::Phase3::GenericCSTAp3, 23
- EncodeSetAgentStateRequest

Com::Objsys::Csta::Phase3::GenericCSTAp3, 23
 EncodeSetDisplayRequest
 Com::Objsys::Csta::Devices::PanasonicKXTDE, 44
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 24
 EncodeSetOrClearDNDRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 24
 EncodeSetOrClearMWRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 24
 EncodeSetRingerStatusRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 25
 EncodeSingleStepTransferRequest
 Com::Objsys::Csta::Devices::SiemensHipath3000p3, 63
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 25
 EncodeSnapshotCallRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 26
 EncodeSnapshotDeviceRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 26
 EncodeStartCDRTransRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 26
 EncodeStartDataPathRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 27
 EncodeStopCDRTransRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 27
 EncodeStopDataPathRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 27
 EncodeTransferCallRequest
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 28
 Encoding
 Com::Objsys::Csta::Phase3::Constants, 6
 ExceptionCallback
 Com::Objsys::Csta::Phase3::PBXSession, 52

 GenericCSTAp3
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 13
 GetAgentState
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 28
 GetDeviceData
 Com::Objsys::Csta::Devices::PanasonicKXTDE, 44
 GetDoNotDisturb
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 29
 GetGroupMembers
 Com::Objsys::Csta::Devices::PanasonicKXTDE, 45
 GetLogicalDevInfo
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 29
 GetPhysicalDevInfo
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 29
 GetSFDevices
 Com::Objsys::Csta::Devices::PanasonicKXTDE, 45
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 29

 HeldDevice
 Com::Objsys::Csta::Phase3::TransferCallInfo, 67
 HeldDeviceCallID
 Com::Objsys::Csta::Phase3::TransferCallInfo, 67
 HeldDeviceNumber
 Com::Objsys::Csta::Phase3::TransferCallInfo, 67
 HoldCall
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 30

 IETF_CSTAp3
 Com::Objsys::Csta::Phase3::IETF_CSTAp3, 38

 LoggingEnabled
 Com::Objsys::Csta::Phase3::PBXSessionHelper, 55
 LoggingFolder
 Com::Objsys::Csta::Phase3::PBXSessionHelper, 55

 MakeACSEAssociation
 Com::Objsys::Csta::Devices::SiemensCap, 60
 Com::Objsys::Csta::Devices::SiemensHipath3000p3, 63
 Com::Objsys::Csta::Devices::UnifyOpenscape4000BER, 69
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 30
 MakeCall
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 30
 MAX_LOGFILE_SIZE
 Com::Objsys::Csta::Phase3::Constants, 6
 MaxLogFileSize
 Com::Objsys::Csta::Phase3::PBXSessionHelper, 55
 MaxReceiveTimeout
 Com::Objsys::Csta::Phase3::PBXSession, 52
 MessageEncoding
 Com::Objsys::Csta::Phase3::PBXSession, 53
 MonitorStart
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 30, 31
 MonitorStop
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 31
 MonitorStopAtNumericDevice
 Com::Objsys::Csta::Phase3::GenericCSTAp3, 32

 Opcodes
 Com::Objsys::Csta::Phase3::Phase3Opcodes, 57
 Open
 Com::Objsys::Csta::Phase3::PBXSession, 49

 PanasonicKXTDA
 Com::Objsys::Csta::Devices::PanasonicKXTDA, 41
 PanasonicKXTDE
 Com::Objsys::Csta::Devices::PanasonicKXTDE, 43
 PanasonicNCP
 Com::Objsys::Csta::Devices::PanasonicNCP, 47
 PBXModels
 Com::Objsys::Csta::Phase3::Constants, 6
 PBXSession
 Com::Objsys::Csta::Phase3::PBXSession, 49
 PBXSystem

Com::Objsys::Csta::Phase3::PBXSession, 53	Com::Objsys::Csta::Phase3::GenericCSTAp3, 34
PDFStart	SetMessageWaiting
Com::Objsys::Csta::Devices::PanasonicKXTDE, 45	Com::Objsys::Csta::Devices::PanasonicKXTDE, 46
PDFStop	Com::Objsys::Csta::Phase3::GenericCSTAp3, 34
Com::Objsys::Csta::Devices::PanasonicKXTDE, 45	SiemensCap
Port	Com::Objsys::Csta::Devices::SiemensCap, 59
Com::Objsys::Csta::Phase3::PBXSession, 53	SiemensHipath3000p3
	Com::Objsys::Csta::Devices::SiemensHipath3000p3, 62
ReadBuffer	SiemensHipath4000
Com::Objsys::Csta::Phase3::SocketState, 66	Com::Objsys::Csta::Devices::SiemensHipath4000, 64
ReadBuffers	
Com::Objsys::Csta::Phase3::SocketState, 66	SingleStepTransfer
ReleaseACSEAssociation	Com::Objsys::Csta::Phase3::GenericCSTAp3, 34 , 35
Com::Objsys::Csta::Phase3::GenericCSTAp3, 32	
ReleaseControlRight	SnapshotCall
Com::Objsys::Csta::Devices::PanasonicKXTDE, 45	Com::Objsys::Csta::Phase3::GenericCSTAp3, 35
RequestSystemStatus	SnapshotDevice
Com::Objsys::Csta::Phase3::GenericCSTAp3, 32	Com::Objsys::Csta::Phase3::GenericCSTAp3, 35
ResetDisplay	StartCDRTransmission
Com::Objsys::Csta::Devices::PanasonicKXTDE, 46	Com::Objsys::Csta::Phase3::GenericCSTAp3, 35
ResponseFromPBX	StartDataPath
Com::Objsys::Csta::Phase3::CSTAContext, 7	Com::Objsys::Csta::Phase3::GenericCSTAp3, 36
Com::Objsys::Csta::Phase3::CSTARResponseInfo, 8	StatusCode
ResponsesFromPBX	Com::Objsys::Csta::Phase3::CSTARResponseInfo, 8
Com::Objsys::Csta::Phase3::CSTAContext, 7	StatusMessage
Com::Objsys::Csta::Phase3::CSTARResponseInfo, 8	Com::Objsys::Csta::Phase3::CSTARResponseInfo, 9
ResponsesFromUA	StopCDRTransmission
Com::Objsys::Csta::Phase3::CSTARResponseInfo, 8	Com::Objsys::Csta::Phase3::GenericCSTAp3, 36
RetrieveCall	StopDataPath
Com::Objsys::Csta::Phase3::GenericCSTAp3, 32	Com::Objsys::Csta::Phase3::GenericCSTAp3, 36
RingDevice	StopRing
Com::Objsys::Csta::Phase3::GenericCSTAp3, 32	Com::Objsys::Csta::Phase3::GenericCSTAp3, 36
	SystemStatusCallback
SendACSEMessage	Com::Objsys::Csta::Phase3::PBXSession, 53
Com::Objsys::Csta::Phase3::PBXSession, 50	
SendData	ThreadContext
Com::Objsys::Csta::Phase3::GenericCSTAp3, 33	Com::Objsys::Csta::Phase3::GenericCSTAp3, 37
SendKmeMessage	TotalLength
Com::Objsys::Csta::Devices::PanasonicKXTDE, 46	Com::Objsys::Csta::Phase3::SocketState, 66
SendMessage	TransferCall
Com::Objsys::Csta::Phase3::PBXSession, 50	Com::Objsys::Csta::Phase3::GenericCSTAp3, 37
SendStoredCDR	TransferFromDevice
Com::Objsys::Csta::Phase3::GenericCSTAp3, 33	Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 65
SendXMLMessage	TransferringCallID
Com::Objsys::Csta::Phase3::PBXSession, 50 , 51	Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 65
SendXMLSession	TransferToDevice
Com::Objsys::Csta::Phase3::PBXSession, 51	Com::Objsys::Csta::Phase3::SingleStepTransferInfo, 65
SessionObject	
Com::Objsys::Csta::Phase3::GenericCSTAp3, 37	UnifyOpenscape4000BER
SetAgentState	Com::Objsys::Csta::Devices::UnifyOpenscape4000BER, 68
Com::Objsys::Csta::Phase3::GenericCSTAp3, 33	
SetDisplay	
Com::Objsys::Csta::Phase3::GenericCSTAp3, 34	
SetDoNotDisturb	

UnifyOpenscapeX5
 Com::Objsys::Csta::Devices::UnifyOpenscapeX5,
 70

WaitForROSEResponse
 Com::Objsys::Csta::Phase3::PBXSession, 51

WaitForXMLResponse
 Com::Objsys::Csta::Phase3::PBXSession, 51

XMLAsyncCallback
 Com::Objsys::Csta::Phase3::PBXSession, 51

XMLCDRCallback
 Com::Objsys::Csta::Phase3::PBXSession, 53

XMLClientCallback
 Com::Objsys::Csta::Phase3::PBXSession, 53

XMLResponseFromPBX
 Com::Objsys::Csta::Phase3::CSTAContext, 7
 Com::Objsys::Csta::Phase3::CSTARResponseInfo, 9

XMLResponseFromUA
 Com::Objsys::Csta::Phase3::CSTARResponseInfo, 9

XMLResponsesFromPBX
 Com::Objsys::Csta::Phase3::CSTAContext, 7
 Com::Objsys::Csta::Phase3::CSTARResponseInfo, 9

XMLSessionMessageTypes
 Com::Objsys::Csta::Phase3::Constants, 6

XMLSystemStatusCallback
 Com::Objsys::Csta::Phase3::PBXSession, 53