CSTA

CSTADLL
Version 1.3.3
CSTADLL
Reference Manual

Objective Systems, Inc. version 1.3.3 May 2013

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.

Author's Contact Information

Comments, suggestions, and inquiries regarding CSTADLL may be submitted via electronic mail to info@obj-sys.com.

Contents

Chapter 1

CSTADLL

The CSTADLL product is a Microsoft .NET 2.0 DLL that allows client code to use a single library to communicate with a PBX device, regardless of what CSTA phase that device uses.

The DLL uses the following namespaces:

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

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

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

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

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

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

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

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

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

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

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

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

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

Chapter 2

Namespace Documentation

2.1 Package Com.Objsys.Csta.Common

Classes

- class CSTAResponseInfo
- class LicenseException
- class PBXSessionException
- class PBXSessionHelper
- class SocketState

2.1.1 Detailed Description

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

2.2 Package Com.Objsys.Csta.Devices

Classes

- class Alcatel4400
- class AlcatelOXO
- class AlcatelOXOp1
- class PanasonicKXTDA
- class PanasonicKXTDE
- class PanasonicNCP
- class SiemensCap
- class SiemensHicom300
- class SiemensHipath3000p2
- class SiemensHipath3000p3
- class SiemensHipath4000
- class SiemensRealitis
- class TadiranCoral

2.2.1 Detailed Description

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

2.3 Package Com.Objsys.Csta.Phase1

Classes

- class GenericCSTAp1
- class IETF_CSTAp1
- class Phase1Opcodes

2.3.1 Detailed Description

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

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

2.4 Package Com.Objsys.Csta.Phase2

Classes

- class GenericCSTAp2
- class IETF_CSTAp2
- class Phase2Opcodes

2.4.1 Detailed Description

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

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

2.5 Package Com.Objsys.Csta.Phase3

Classes

- class GenericCSTAp3
- class IETF_CSTAp3
- class Phase3Opcodes
- class SingleStepTransferInfo
- class TransferCallInfo

2.5.1 Detailed Description

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

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

Chapter 3

Class Documentation

3.1 Alcatel4400 Class Reference

Inherits Com::Objsys::Csta::Phase2::IETF_CSTAp2.

Public Member Functions

- Alcatel4400 ()
- override CSTAResponseInfo MakeACSEAssociation ()

3.1.1 Detailed Description

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

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Alcatel4400 ()

Default constructor.

3.1.3 Member Function Documentation

3.1.3.1 override CSTAResponseInfo MakeACSEAssociation () `[virtual]`

Establish an ACSE association with the PBX.

Returns

A CSTAResponseInfo object.

Reimplemented from GenericCSTAp2.

3.2 AlcatelOXO Class Reference

Inherits Com::Objsys::Csta::Phase2::IETF_CSTAp2.

Public Member Functions

- AlcatelOXO ()
- override CSTAResponseInfo MakeACSEAssociation ()

3.2.1 Detailed Description

Implements CSTA phase 2 operations for the Alcatel OXO device.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 AlcatelOXO ()

Default constructor.

3.2.3 Member Function Documentation

3.2.3.1 override CSTAResponseInfo MakeACSEAssociation () `[virtual]`

Establish an ACSE association with the PBX.

Returns

A CSTAResponseInfo object.

Reimplemented from GenericCSTAp2.

3.3 AlcatelOXOp1 Class Reference

Inherits Com::Objsys::Csta::Phase1::IETF_CSTAp1.

Public Member Functions

- AlcatelOXOp1 ()
- override CSTAResponseInfo MakeACSEAssociation ()

3.3.1 Detailed Description

Implements CSTA phase 1 operations for the Alcatel OXO device.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 AlcatelOXOp1 ()

Default constructor.

3.3.3 Member Function Documentation

3.3.3.1 override CSTAResponseInfo MakeACSEAssociation () `[virtual]`

Establish an ACSE association with the PBX.

Returns

A CSTAResponseInfo object.

Reimplemented from GenericCSTAp1.

3.4 CSTAResponseInfo Class Reference

Properties

- byte[ ] ResponseFromPBX `[get, set]`
- byte[ ][ ] ResponsesFromPBX `[get, set]`
- int StatusCode `[get, set]`
- string StatusMessage `[get, set]`

3.4.1 Detailed Description

Contains information about a PBX operation that was attempted.

3.4.2 Property Documentation

3.4.2.1 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.

3.4.2.2 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.

3.4.2.3 int StatusCode `[get, set]`

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

3.4.2.4 string StatusMessage `[get, set]`

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

3.5 GenericCSTAp1 Class Reference

Inherited by SiemensHicom300, and IETF_CSTAp1.

Public Member Functions

- virtual CSTAResponseInfo AnswerCall (string deviceToLift)
- virtual CSTAResponseInfo DivertCall (string divertFrom, string divertTo)
- int EncodeROSERequestHeader (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, Phase1Opcodes.Opcodes opcode, out InvokeId savedInvokeId)
- GenericCSTAp1 ()
- virtual CSTAResponseInfo MakeACSEAssociation ()
- virtual CSTAResponseInfo MakeCall (string callingDevice, string calledDevice)
- virtual CSTAResponseInfo MonitorStart (string deviceToMonitor)
- virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID crossRefId)
- virtual CSTAResponseInfo MonitorStop (int crossRefId)
- virtual CSTAResponseInfo QueryDevice (string deviceToQuery)
- virtual CSTAResponseInfo TransferCall (string heldDevice, string connectedDevice)

3.5.1 Detailed Description

Implements CSTA phase 1 operations using BER.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 GenericCSTAp1 ()

Default constructor. Indicates that phase 1 is in effect.

3.5.3 Member Function Documentation

3.5.3.1 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.

3.5.3.2 virtual CSTAResponseInfo DivertCall (string *divertFrom*, string *divertTo*) `[virtual]`

Diverts a call from a source to a destination.

Parameters

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

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

Returns

A CSTAResponseInfo object.

3.5.3.3 int EncodeROSERequestHeader (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, Phase1Opcodes.Opcodes *opcode*, out InvokeId *savedInvokeId*)

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

Parameters

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

encodeBuffer An Asn1BerEncodeBuffer instance containing the already encoded CSTA message.

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

savedInvokeId An InvokeId object variable that will get populated with the reference to the InvokeId object that gets encoded into the ROSE header. This is used when the response is received to match the response with the request.

Returns

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

3.5.3.4 virtual CSTAResponseInfo MakeACSEAssociation () `[virtual]`

Establish an ACSE association with the PBX.

Returns

A CSTAResponseInfo object.

Reimplemented in AlcatelOXOp1.

3.5.3.5 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.

3.5.3.6 virtual CSTAResponseInfo MonitorStart (string *deviceToMonitor*) `[virtual]`

Issues a MonitorStart request to the PBX.

Parameters

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

Returns

A CSTAResponseInfo object.

3.5.3.7 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.

3.5.3.8 virtual CSTAResponseInfo MonitorStop (int *crossRefId*) `[virtual]`

Stop a previously started PBX monitor request.

Parameters

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

Returns

A CSTAResponseInfo object.

3.5.3.9 virtual CSTAResponseInfo QueryDevice (string *deviceToQuery*) `[virtual]`

Queries a device.

Parameters

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

Returns

A CSTAResponseInfo object.

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

Transfers a call from one device to another.

Parameters

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

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

Returns

A CSTAResponseInfo object.

3.6 GenericCSTAp2 Class Reference

Inherited by SiemensHipath3000p2, and IETF_CSTAp2.

Public Member Functions

- virtual CSTAResponseInfo AnswerCall (string deviceToLift)
- virtual CSTAResponseInfo DivertCall (string divertFrom, string divertTo)
- int EncodeROSERequestHeader (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, Phase2Opcodes.Opcodes opcode, out InvokeId savedInvokeId)
- GenericCSTAp2 ()
- virtual CSTAResponseInfo MakeACSEAssociation ()
- virtual CSTAResponseInfo MakeCall (string callingDevice, string calledDevice)
- virtual CSTAResponseInfo MonitorStart (string deviceToMonitor)
- virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID crossRefId)
- virtual CSTAResponseInfo MonitorStop (int crossRefId)
- virtual CSTAResponseInfo QueryDevice (string deviceToQuery)
- virtual CSTAResponseInfo TransferCall (string heldDevice, string connectedDevice)

3.6.1 Detailed Description

Implements CSTA phase 2 operations using BER.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 GenericCSTAp2 ()

Default constructor. Indicates that phase 2 is in effect.

3.6.3 Member Function Documentation

3.6.3.1 virtual CSTAResponseInfo AnswerCall (string *deviceToLift*) `[virtual]`

Answers a call.

Parameters

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

Returns

A CSTAResponseInfo object.

3.6.3.2 virtual CSTAResponseInfo DivertCall (string *divertFrom*, string *divertTo*) `[virtual]`

Diverts a call from a source to a destination.

Parameters

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

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

Returns

A CSTAResponseInfo object.

3.6.3.3 int EncodeROSERequestHeader (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, Phase2Opcodes.Opcodes *opcode*, out InvokeId *savedInvokeId*)

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

Parameters

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

encodeBuffer An Asn1BerEncodeBuffer instance containing the already encoded CSTA message.

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

savedInvokeId An InvokeId object variable that will get populated with the reference to the InvokeId object that gets encoded into the ROSE header. This is used when the response is received to match the response with the request.

Returns

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

3.6.3.4 virtual CSTAResponseInfo MakeACSEAssociation () `[virtual]`

Establish an ACSE association with the PBX.

Returns

A CSTAResponseInfo object.

Reimplemented in Alcatel4400, and AlcatelOXO.

3.6.3.5 virtual CSTAResponseInfo MakeCall (string *callingDevice*, string *calledDevice*) `[virtual]`

Instruct the PBX to place a call.

Parameters

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

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

Returns

A CSTAResponseInfo object.

3.6.3.6 virtual CSTAResponseInfo MonitorStart (string *deviceToMonitor*) `[virtual]`

Issues a MonitorStart request to the PBX.

Parameters

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

Returns

A CSTAResponseInfo object.

3.6.3.7 virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID *crossRefId*) `[virtual]`

Stop a previously started PBX monitor request.

Parameters

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

Returns

A CSTAResponseInfo object.

3.6.3.8 virtual CSTAResponseInfo MonitorStop (int *crossRefId*) `[virtual]`

Stop a previously started PBX monitor request.

Parameters

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

Returns

A CSTAResponseInfo object.

3.6.3.9 virtual CSTAResponseInfo QueryDevice (string *deviceToQuery*) `[virtual]`

Queries a device.

Parameters

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

Returns

A CSTAResponseInfo object.

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

Transfers a call from one device to another.

Parameters

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

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

Returns

A CSTAResponseInfo object.

3.7 GenericCSTAp3 Class Reference

Inherited by SiemensHipath3000p3, and IETF_CSTAp3.

Public Member Functions

- virtual CSTAResponseInfo AnswerCall (ConnectionID callToAnswer, string deviceToLift)
- virtual CSTAResponseInfo AnswerCall (string deviceToLift)
- virtual CSTAResponseInfo ClearMessageWaiting (string targetDevice)
- virtual CSTAResponseInfo ConsultationCall (ConnectionID existingCall, string targetDevice)
- virtual int EncodeROSERequestHeader (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, Phase3Opcodes.Opcodes opcode, out InvokeId savedInvokeId)
- GenericCSTAp3 ()
- virtual CSTAResponseInfo GetSFDevices (ReqDeviceCategory deviceCategory)
- virtual CSTAResponseInfo GetSFDevices ()
- virtual CSTAResponseInfo MakeACSEAssociation ()
- virtual CSTAResponseInfo MakeCall (string callingDevice, string calledDevice)
- virtual CSTAResponseInfo MonitorStart (string deviceToMonitor)
- virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID crossRefId)
- virtual CSTAResponseInfo MonitorStop (int crossRefId)
- virtual CSTAResponseInfo ReleaseACSEAssociation ()
- virtual CSTAResponseInfo RingDevice (string targetDevice, string targetRinger, long ringPattern)
- virtual CSTAResponseInfo SendData (IOCrossRefID ioCrossRef, string text)
- virtual CSTAResponseInfo SetDisplay (string targetDevice, string text)
- virtual CSTAResponseInfo SetMessageWaiting (string targetDevice)
- virtual CSTAResponseInfo SingleStepTransfer (SingleStepTransferInfo sstInfo)
- virtual CSTAResponseInfo StartDataPath (string targetDevice)
- virtual CSTAResponseInfo StopDataPath (IOCrossRefID ioCrossRef)
- virtual CSTAResponseInfo StopRing (string targetDevice, string targetRinger, long ringPattern)
- virtual CSTAResponseInfo TransferCall (TransferCallInfo tcInfo)
- virtual CSTAResponseInfo TransferCall (ConnectionID existingCall, string targetDevice)
- virtual CSTAResponseInfo TransferCallAfterConsult (ConnectionID initiatedCall, ConnectionID originalCall)

3.7.1 Detailed Description

Implements CSTA phase 3 operations using BER.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 GenericCSTAp3 ()

Default constructor. Indicates that phase 3 is in effect.

3.7.3 Member Function Documentation

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

Answers a call.

Parameters

callToAnswer ConnectionID of an existing call (such as initiated through MakeCall()).

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

Returns

A CSTAResponseInfo object.

3.7.3.2 virtual CSTAResponseInfo AnswerCall (string *deviceToLift*) [virtual]

Answers a call.

Parameters

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

Returns

A CSTAResponseInfo object.

3.7.3.3 virtual CSTAResponseInfo ClearMessageWaiting (string *targetDevice*) [virtual]

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

Parameters

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

Returns

A CSTAResponseInfo object.

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

Instruct the PBX to do a consultation call.

Parameters

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

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

Returns

A CSTAResponseInfo object.

3.7.3.5 virtual int EncodeROSERequestHeader (CSTAResponseInfo *response*, Asn1BerEncodeBuffer *encodeBuffer*, Phase3Opcodes.Opcodes *opcode*, out InvokeId *savedInvokeId*) `[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.

savedInvokeId An InvokeId object variable that will get populated with the reference to the InvokeId object that gets encoded into the ROSE header. This is used when the response is received to match the response with the request.

Returns

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

3.7.3.6 virtual CSTAResponseInfo GetSFDevices (ReqDeviceCategory *deviceCategory*) `[virtual]`

Sends a Get Switching Function Devices request to the PBX.

Parameters

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

Returns

A CSTAResponseInfo object.

3.7.3.7 virtual CSTAResponseInfo GetSFDevices () `[virtual]`

Sends a Get Switching Function Devices request to the PBX.

Returns

A CSTAResponseInfo object.

Reimplemented in PanasonicKXTDE.

3.7.3.8 virtual CSTAResponseInfo MakeACSEAssociation () `[virtual]`

Establish an ACSE association with the PBX.

Returns

A CSTAResponseInfo object.

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

Instruct the PBX to place a call.

Parameters

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

Returns

A CSTAResponseInfo object.

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

Issues a MonitorStart request to the PBX.

Parameters

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

Returns

A CSTAResponseInfo object.

3.7.3.11 virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID *crossRefId*) [virtual]

Stop a previously started PBX monitor request.

Parameters

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

Returns

A CSTAResponseInfo object.

3.7.3.12 virtual CSTAResponseInfo MonitorStop (int *crossRefId*) [virtual]

Stop a previously started PBX monitor request.

Parameters

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

Returns

A CSTAResponseInfo object.

3.7.3.13 virtual CSTAResponseInfo ReleaseACSEAssociation () [virtual]

Releases an ACSE association with a PBX device.

Returns

A CSTAResponseInfo object.

3.7.3.14 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, "'000000010000101000000101'B").

ringPattern The indicator of the ring pattern to use.

Returns

A CSTAResponseInfo object.

3.7.3.15 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.

3.7.3.16 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.

3.7.3.17 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.

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

Perform a single step transfer.

Parameters

sstInfo A SingleStepTransferInfo object.

Returns

A CSTAResponseInfo object.

3.7.3.19 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.

3.7.3.20 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.

3.7.3.21 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, "'00000001000010100000101'B").

ringPattern The indicator of the ring pattern to stop.

Returns

A CSTAResponseInfo object.

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

Transfers a call from one device to another.

Parameters

tcInfo A TransferCallInfo object.

Returns

A CSTAResponseInfo object.

3.7.3.23 virtual CSTAResponseInfo TransferCall (ConnectionID *existingCall*, string *targetDevice*) [virtual]

Transfers a call from one device to another.

Parameters

existingCall ConnectionID of an existing call (such as initiated through MakeCall()).

targetDevice The device (e.g., "800") to which the call is to be transferred.

Returns

A CSTAResponseInfo object.

3.7.3.24 virtual CSTAResponseInfo TransferCallAfterConsult (ConnectionID *initiatedCall*, ConnectionID *originalCall*) [virtual]

Transfers a call after a consultation call has been done.

Parameters

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

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

Returns

A CSTAResponseInfo object.

3.8 IETF_CSTAp1 Class Reference

Inherits Com::Objsys::Csta::Phase1::GenericCSTAp1.

Inherited by AlcatelOXOp1, SiemensRealitis, and TadiranCoral.

Public Member Functions

- IETF_CSTAp1 ()

3.8.1 Detailed Description

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

3.8.2 Constructor & Destructor Documentation

3.8.2.1 IETF_CSTAp1 ()

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

3.9 IETF_CSTAp2 Class Reference

Inherits Com::Objsys::Csta::Phase2::GenericCSTAp2.

Inherited by Alcatel4400, and AlcatelOXO.

Public Member Functions

- IETF_CSTAp2 ()

3.9.1 Detailed Description

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

3.9.2 Constructor & Destructor Documentation

3.9.2.1 IETF_CSTAp2 ()

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

3.10 IETF_CSTAp3 Class Reference

Inherits Com::Objsys::Csta::Phase3::GenericCSTAp3.

Inherited by PanasonicKXTDE, and SiemensCap.

Public Member Functions

- IETF_CSTAp3 ()

3.10.1 Detailed Description

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

3.10.2 Constructor & Destructor Documentation

3.10.2.1 IETF_CSTAp3 ()

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

3.11 LicenseException Class Reference

3.11.1 Detailed Description

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

3.12 PanasonicKXTDA Class Reference

Inherits Com::Objsys::Csta::Devices::PanasonicKXTDE.

3.12.1 Detailed Description

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

3.13 PanasonicKXTDE Class Reference

Inherits Com::Objsys::Csta::Phase3::IETF_CSTAp3.

Inherited by PanasonicKXTDA, and PanasonicNCP.

Public Member Functions

- CSTAResponseInfo AcquireControlRight (string targetDevice)
- CSTAResponseInfo ClearMessageWaiting (string originatingDevice, string targetDevice)
- override CSTAResponseInfo GetSFDevices ()
- CSTAResponseInfo ReleaseControlRight (string targetDevice)
- CSTAResponseInfo SetMessageWaiting (string originatingDevice, string targetDevice)

3.13.1 Detailed Description

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

3.13.2 Member Function Documentation

3.13.2.1 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.

3.13.2.2 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.

3.13.2.3 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.

3.13.2.4 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.

3.13.2.5 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.

3.14 PanasonicNCP Class Reference

Inherits Com::Objsys::Csta::Devices::PanasonicKXTDE.

3.14.1 Detailed Description

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

3.15 PBXSessionException Class Reference

3.15.1 Detailed Description

Defines an exception that occurs while communicating with a PBX.

3.16 PBXSessionHelper Class Reference

Public Member Functions

- delegate void AsyncCallback (byte[ ] asyncData)

Static Public Member Functions

- static void Close ()
- static void Init (string pbxSystem, int port)
- static void Open ()
- static void SendMessage (string messageType, byte[ ] message, int messageLength, Asn1Choice invokeId)
- static void SendMessage (byte[ ] message, int messageLength, Asn1Choice invokeId)
- static SocketState WaitForACSEResponse ()
- static SocketState WaitForROSEResponse (Asn1Choice invokeId)

Properties

- static AsyncCallback ClientCallback `[get, set]`
- static bool DebugMode `[get, set]`
- static bool LoggingEnabled `[get, set]`
- static string LoggingFolder `[get, set]`

3.16.1 Detailed Description

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

3.16.2 Member Function Documentation

3.16.2.1 delegate void AsyncCallback (byte[ ] *asyncData*)

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

Parameters

asyncData The data received asynchronously from the PBX.

3.16.2.2 static void Close () `[static]`

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

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

Initializes the PBX Session.

Parameters

pbxSystem The name or IP address of the PBX system.

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

3.16.2.4 static void Open () [static]

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

3.16.2.5 static void SendMessage (string *messageType*, byte[ ] *message*, int *messageLength*, Asn1Choice *invokeId*) [static]

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

Parameters

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

message Byte array containing the encoded message to send.

messageLength The length of the encoded message.

invokeId The invoke id object from the message's ROSE header. This parameter is cast up to Asn1Choice since this method is phase-independent, and the InvokeId classes for all three phases derive from the Asn1Choice class in the ASN1C C# runtime.

3.16.2.6 static void SendMessage (byte[ ] *message*, int *messageLength*, Asn1Choice *invokeId*) [static]

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

Parameters

message Byte array containing the encoded message to send.

messageLength The length of the encoded message.

invokeId The invoke id object from the message's ROSE header. This parameter is cast up to Asn1Choice since this method is phase-independent, and the InvokeId classes for all three phases derive from the Asn1Choice class in the ASN1C C# runtime.

3.16.2.7 static SocketState WaitForACSEResponse () [static]

This method waits for a response to an ACSE message. ACSE messages do not have the ROSE header.

Returns

A SocketState instance that contains the response that comes back from the PBX.

3.16.2.8 static SocketState WaitForROSEResponse (Asn1Choice *invokeId*) `[static]`

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

Parameters

invokeId The invoke id object that was encoded into the ROSE header. This is used to match received messages back to the sending message. The parameter is cast up to Asn1Choice because this method is phase-independent, and the InvokeId classes for all three phases derive from the Asn1Choice class in the ASN1C C# runtime.

Returns

A SocketState instance that contains the response that comes back from the PBX.

3.16.3 Property Documentation

3.16.3.1 AsyncCallback ClientCallback `[static, get, set]`

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

3.16.3.2 bool DebugMode `[static, get, set]`

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

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

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

3.16.3.4 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.

3.17 Phase1Opcodes Class Reference

Public Types

- enum Opcodes

3.17.1 Detailed Description

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

3.17.2 Member Enumeration Documentation

3.17.2.1 enum Opcodes

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

3.18 Phase2Opcodes Class Reference

Public Types

- enum Opcodes

3.18.1 Detailed Description

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

3.18.2 Member Enumeration Documentation

3.18.2.1 enum Opcodes

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

3.19 Phase3Opcodes Class Reference

Public Types

- enum Opcodes

3.19.1 Detailed Description

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

3.19.2 Member Enumeration Documentation

3.19.2.1 enum Opcodes

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

3.20 SiemensCap Class Reference

Inherits Com::Objsys::Csta::Phase3::IETF_CSTAp3.

Inherited by SiemensHipath4000.

3.20.1 Detailed Description

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

3.21 SiemensHicom300 Class Reference

Inherits Com::Objsys::Csta::Phase1::GenericCSTAp1.

Public Member Functions

- SiemensHicom300 ()

3.21.1 Detailed Description

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

3.21.2 Constructor & Destructor Documentation

3.21.2.1 SiemensHicom300 ()

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

3.22 SiemensHipath3000p2 Class Reference

Inherits Com::Objsys::Csta::Phase2::GenericCSTAp2.

Public Member Functions

- SiemensHipath3000p2 ()

3.22.1 Detailed Description

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

3.22.2 Constructor & Destructor Documentation

3.22.2.1 SiemensHipath3000p2 ()

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

3.23 SiemensHipath3000p3 Class Reference

Inherits Com::Objsys::Csta::Phase3::GenericCSTAp3.

Public Member Functions

- SiemensHipath3000p3 ()

3.23.1 Detailed Description

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

3.23.2 Constructor & Destructor Documentation

3.23.2.1 SiemensHipath3000p3 ()

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

3.24 SiemensHipath4000 Class Reference

Inherits Com::Objsys::Csta::Devices::SiemensCap.

3.24.1 Detailed Description

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

3.25 SiemensRealitis Class Reference

Inherits Com::Objsys::Csta::Phase1::IETF_CSTAp1.

3.25.1 Detailed Description

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

3.26 SingleStepTransferInfo Class Reference

Properties

- string TransferFromDevice `[get, set]`
- string TransferringCallID `[get, set]`
- string TransferToDevice `[get, set]`

3.26.1 Detailed Description

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

3.26.2 Property Documentation

3.26.2.1 string TransferFromDevice `[get, set]`

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

3.26.2.2 string TransferringCallID `[get, set]`

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

3.26.2.3 string TransferToDevice `[get, set]`

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

3.27 SocketState Class Reference

Properties

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

3.27.1 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.

3.27.2 Property Documentation

3.27.2.1 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.

3.27.2.2 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.

3.27.2.3 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.

3.27.2.4 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.

3.28 TadiranCoral Class Reference

Inherits Com::Objsys::Csta::Phase1::IETF_CSTAp1.

3.28.1 Detailed Description

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

3.29 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]`

3.29.1 Detailed Description

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

3.29.2 Property Documentation

3.29.2.1 string ActiveDeviceCallID `[get, set]`

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

3.29.2.2 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.

3.29.2.3 string ConnectedDevice `[get, set]`

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

3.29.2.4 string HeldDevice `[get, set]`

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

3.29.2.5 string HeldDeviceCallID `[get, set]`

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

3.29.2.6 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.

Index