CSTADLL

Version 2.5.0

Objective Systems, Inc.

October 2023

CSTADLL

Author's Contact Information. Comments, suggestions, and inquiries regarding CSTADLL or this document may be sent by electronic mail to <info@obj-sys.com>.

Table of Contents

1. Namespace Documentation 1
    Com 1
    Com::Objsys 1
    Com::Objsys::Asn1::Runtime 1
    Com::Objsys::Csta 1
    Com::Objsys::Csta::Devices 1
        Classes 1
    Com::Objsys::Csta::Phase2 2
        Classes 2
    System 3
    System::Collections::Generic 3
    System::IO 3
    System::Linq 3
    System::Net 3
    System::Net::Sockets 3
    System::Reflection 3
    System::Runtime::InteropServices 3
    System::Text 3
    System::Threading 3
    System::Threading::Tasks 3
2. CSTABERPhase2 4
3. Class Documentation 6
    Com::Objsys::Csta::Devices::Alcatel4400 class Reference 6
        6
        6
        Alcatel4400 (string pbxSystem, int port) 6
        Alcatel4400 (PBXSession sessionObject) 6
        override CSTAResponseInfo MakeACSEAssociation () 6
        override int EncodeACSEConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer) 7
        override int EncodeMakeCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice) 7
    Com::Objsys::Csta::Devices::AlcatelOXE class Reference 7
        7
        AlcatelOXE (string pbxSystem, int port) 8
        AlcatelOXE (PBXSession sessionObject) 8
    ApplicationException class Reference 8
    Com::Objsys::Csta::Phase2::Constants class Reference 8
        8
        9
        Public Attributes 9
        enum ACSEMessageTypes 10
        enum CallbackInvocationMechanisms 10
        enum CommunicationTypes 11
        enum Encoding 11
        enum PBXModels 12
        enum XMLSessionMessageTypes 13
        Member Data Documentation 15
    Com::Objsys::Csta::Phase2::CSTAContext class Reference 16
        16
        16

Private Attributes 17
..... 18
..... 18
..... 18
CSTAContext () 18
~CSTAContext () 18
static CSTAContext Instance () 18
Com::Objsys::Csta::Phase2::CSTAEncDec class Reference 18
..... 18
..... 19
..... 19
..... 19
Com::Objsys::Csta::Phase2::CSTAResponseInfo class Reference 19
Private Attributes 19
..... 19
Com::Objsys::Csta::Phase2::GenericCSTAp2 class Reference 20
Protected Attributes 20
..... 20
..... 20
..... 21
..... 22
..... 22
virtual CSTAResponseInfo AnswerCall (string deviceToLift) 23
virtual CSTAResponseInfo AnswerCall (ConnectionID callToAnswer) 23
virtual CSTAResponseInfo AnswerCall (ConnectionID callToAnswer, string deviceToLift) 23
virtual CSTAResponseInfo ClearConnection (ConnectionID connToClear) 23
virtual CSTAResponseInfo ClearDoNotDisturb (string targetDevice) 24
virtual CSTAResponseInfo ClearMessageWaiting (string targetDevice) 24
virtual CSTAResponseInfo ConferenceCall (ConnectionID heldCall, ConnectionID activeCall)..... 24
virtual CSTAResponseInfo ConsultationCall (ConnectionID existingCall, string targetDevice) 24
virtual CSTAResponseInfo DivertCall (string divertFrom, string divertTo) 25
int EncodeROSERequestHeader (CSTAResponseInfo response, Asn1BerEncodeBuffer encode-Buffer, Phase2Opcodes.Opcodes opcode) 25
GenericCSTAp2 (string pbxSystem, int port) 25
GenericCSTAp2 (PBXSession sessionObject) 26
virtual CSTAResponseInfo HoldCall (ConnectionID callToHold) 26
virtual CSTAResponseInfo MakeACSEAssociation () 26
virtual CSTAResponseInfo MakeCall (string callingDevice, string calledDevice) 26
virtual CSTAResponseInfo MonitorStart (string deviceToMonitor) 26
virtual CSTAResponseInfo MonitorStart (ConnectionID callToMonitor) 27
virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID crossRefID) 27
virtual CSTAResponseInfo MonitorStop (string monitoredDevice) 27
virtual CSTAResponseInfo MonitorStop (long monitoredDevice) 28
virtual CSTAResponseInfo QueryDevice (string deviceToQuery) 28
virtual CSTAResponseInfo ReleaseACSEAssociation () 28
virtual CSTAResponseInfo RetrieveCall (ConnectionID callToRetrieve) 28
virtual CSTAResponseInfo SetAgentState (string targetDevice, AgentParameter agentParam) 29
virtual CSTAResponseInfo SetDoNotDisturb (string targetDevice) 29
virtual CSTAResponseInfo SetMessageWaiting (string targetDevice) 29
virtual CSTAResponseInfo SetOrClearForwarding (string fromDevice, ForwardingType fwdType, string toDevice) 29
virtual CSTAResponseInfo SnapshotDevice (string deviceToSnapshot) 30
virtual CSTAResponseInfo TransferCall (ConnectionID initiatedCall, ConnectionID originalCall) 30

virtual CSTAResponseInfo TransferCall (string heldDevice, string connectedDevice) 30
virtual int EncodeACSEConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer) 31
virtual int EncodeACSEReleaseRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer) 31
virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToLift) 31
virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer) 32
virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer, string deviceToLift) 32
virtual int EncodeClearConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID connToClear) 32
virtual int EncodeConferenceCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID heldCall, ConnectionID activeCall) 33
virtual int EncodeConsultationCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID existingCall, string targetDevice) 33
virtual int EncodeDivertCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string divertFrom, string divertTo) 33
virtual int EncodeHoldCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToHold) 34
virtual int EncodeMakeCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice) 34
virtual int EncodeMonitorStartRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToMonitor) 34
virtual int EncodeMonitorStartRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToMonitor) 35
virtual int EncodeMonitorStopRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, MonitorCrossRefID crossRefObj) 35
virtual int EncodeQueryDeviceRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToQuery) 35
virtual int EncodeRetrieveCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToRetrieve) 36
virtual int EncodeSetAgentStateRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, AgentParameter agentParam) 36
virtual int EncodeSetOrClearDNDRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool dndOn) 36
virtual int EncodeSetOrClearFwdRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string fromDevice, ForwardingType fwdType, string toDevice) 37
virtual int EncodeSetOrClearMWRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool indicatorOn) 37
virtual int EncodeSnapshotDeviceRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToSnapshot) 38
virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID initiatedCall, ConnectionID originalCall) 38
virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string heldDevice, string connectedDevice) 38
Com::Objsys::Csta::Phase2::IETF_CSTAp2 class Reference 39
..... 39
IETF_CSTAp2 (string pbxSystem, int port) 39
IETF_CSTAp2 (PBXSession sessionObject) 39
Com::Objsys::Csta::Phase2::LicenseException class Reference 39
..... 39
Com::Objsys::Csta::Phase2::LicenseHelper class Reference 40

..... 40
..... 40
static bool CheckLicenseSettings (LicenseOptions.BERPhases phase) 40
static bool CheckLicenseSettings (LicenseOptions.XMLEditions edition) 40
static void FreeLicense (bool close) 41
static string GenNotEnabledMsg (Constants.Encoding encoding, ushort phaseOrEdition) 41
static void HandleException (PBXSession sessionObject, string text) 41
Com::Objsys::Csta::Phase2::LicenseOptions class Reference 41
..... 41
Private Attributes 42
..... 42
enum BERPhases 42
enum XMLEditions 42
Com::Objsys::Csta::Phase2::PBXSession class Reference 43
Private Attributes 43
..... 44
..... 45
..... 45
..... 46
..... 46
..... 46
delegate void AsyncCallback (PBXSession sessionObject, byte[] asyncData) 46
delegate void AsyncExceptionCallback (PBXSession sessionObject, ApplicationException exception) 47
void Close (CSTAContext threadContext) 47
delegate void ConnectionCallback (PBXSession sessionObject) 47
void Open (CSTAContext threadContext) 47
PBXSession (string pbxSystem, int port) 48
SocketState SendACSEMessage (byte[] message, int messageLength, Constants.ACSEMessageTypes messageType, CSTAContext threadContext) 48
void SendMessage (byte[] message, int messageLength, CSTAContext threadContext) 48
void SendMessage (string messageType, byte[] message, int messageLength, CSTAContext threadContext) 49
void SendXMLMessage (string strMessage, CSTAContext threadContext) 49
void SendXMLMessage (string messageType, string strMessage, CSTAContext threadContext) 49
SocketState SendXMLSession (string strMessage, Constants.XMLSessionMessageTypes enmMessageType, CSTAContext threadContext) 49
void WaitForROSEResponse (CSTAContext threadContext) 50
void WaitForXMLResponse (CSTAContext threadContext) 50
delegate void XMLAsyncCallback (PBXSession sessionObject, string message) 50
static void Init () 50
void AlcatelInit (CSTAContext threadContext) 51
void Connect (CSTAContext threadContext) 51
void SendMessageInternal (byte[] message, int messageLength, CSTAContext threadContext) 51
void WaitForResetSessionResponse (CSTAContext threadContext) 51
void WaitForStopSessionResponse (CSTAContext threadContext) 51
Com::Objsys::Csta::Phase2::PBXSessionException class Reference 52
..... 52
Com::Objsys::Csta::Phase2::PBXSessionHelper class Reference 52
..... 52
..... 52
..... 52
..... 52
Com::Objsys::Csta::Phase2::PBXSessionHelperPhase2 class Reference 53

..... 53
..... 53
Com::Objsys::Csta::Phase2::Phase2Opcodes class Reference 53
..... 53
enum Opcodes 55
Com::Objsys::Csta::Phase2::ROSEParseInfo class Reference 58
..... 58
Com::Objsys::Csta::Devices::SamsungSCM class Reference 58
Private Attributes 58
..... 58
..... 59
..... 59
virtual CSTAResponseInfo LinkSetup (bool returnNumberPlan) 59
sealed override CSTAResponseInfo MakeACSEAssociation () 59
SamsungSCM (string pbxSystem, int port) 59
SamsungSCM (PBXSession sessionObject) 60
virtual int EncodeLinkSetupRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, bool returnNumberPlan) 60
Com::Objsys::Csta::Devices::SiemensHipath3000p2 class Reference 60
..... 60
SiemensHipath3000p2 (string pbxSystem, int port) 60
SiemensHipath3000p2 (PBXSession sessionObject) 61
Com::Objsys::Csta::Phase2::SocketState class Reference 61
Private Attributes 61
..... 62
..... 63
..... 63
void Reset () 64
SocketState () 64
void ValidateBufferLength () 64
4. File Documentation 65
_SeqOfFloatLicProductInfo.cs File Reference 65
Alcatel4400.cs File Reference 65
Classes 65
AlcatelOXE.cs File Reference 65
Classes 65
AlcatelOXO.cs File Reference 66
Constants.cs File Reference 66
Classes 66
CSTAContext.cs File Reference 66
Classes 66
CSTAEncDec.cs File Reference 66
Classes 66
CSTAResponseInfo.cs File Reference 67
Classes 67
FloatLicInfo.cs File Reference 67
FloatLicProductInfo.cs File Reference 67
GenericCSTAp2.cs File Reference 67
Classes 67
IETF_CSTAp2.cs File Reference 67
Classes 67
LicenseBitFlags.cs File Reference 68
LicenseChoice.cs File Reference 68
LicenseChoice_hosts.cs File Reference 68

LicenseData.cs File Reference 68
LicenseData_licProcIds.cs File Reference 68
LicensedProduct.cs File Reference 68
LicenseException.cs File Reference 69
Classes 69
LicenseHelper.cs File Reference 69
Classes 69
LicenseHost.cs File Reference 69
LicenseHost_id.cs File Reference 69
LicenseOptions.cs File Reference 69
Classes 69
LicenseUserInfo.cs File Reference 70
LicenseValidityPeriod.cs File Reference 70
PanasonicKXNS.cs File Reference 70
PanasonicKXTDA.cs File Reference 70
PanasonicKXTDE.cs File Reference 70
PanasonicNCP.cs File Reference 70
PanasonicNXS.cs File Reference 71
PBXSession.cs File Reference 71
Classes 71
PBXSessionException.cs File Reference 71
Classes 71
PBXSessionHelper.cs File Reference 71
Classes 71
PBXSessionHelperEd3.cs File Reference 72
PBXSessionHelperEd4.cs File Reference 72
PBXSessionHelperEd5.cs File Reference 72
PBXSessionHelperEd6.cs File Reference 72
PBXSessionHelperPhase1.cs File Reference 72
PBXSessionHelperPhase2.cs File Reference 72
Classes 72
PBXSessionHelperPhase3.cs File Reference 73
Phase2Opcodes.cs File Reference 73
Classes 73
PhilipsSopho.cs File Reference 73
ResetSessionInfo.cs File Reference 73
ROSEParseInfo.cs File Reference 73
Classes 73
RunTimeFloatLicInfo.cs File Reference 74
RunTimeLicAckResp.cs File Reference 74
RunTimeLicCheckInReq.cs File Reference 74
RunTimeLicCheckOutReq.cs File Reference 74
RunTimeLicCheckOutResp.cs File Reference 74
RunTimeLicPIDUpdateReq.cs File Reference 74
SamsungSCM.cs File Reference 74
Classes 74
SiemensCap.cs File Reference 75
SiemensHicom300.cs File Reference 75
SiemensHipath3000p2.cs File Reference 75
Classes 75
SiemensHipath3000p3.cs File Reference 75
SiemensHipath4000.cs File Reference 75
SiemensRealitis.cs File Reference 76
SocketState.cs File Reference 76

Classes 76
TadiranCoral.cs File Reference 76
uaSIPInvite.cs File Reference 76
UnifyOpenscape4000BER.cs File Reference 76
UnifyOpenscapeVoice.cs File Reference 76
UnifyOpenscapeX5.cs File Reference 77
Version.cs File Reference 77
VodiaSNOMOne.cs File Reference 77
XMLParseInfo.cs File Reference 77

List of Tables

3.1. Parameters 6
3.2. Parameters 6
3.3. Parameters 7
3.4. Parameters 7
3.5. Parameters 8
3.6. Parameters 8
3.7. Parameters 23
3.8. Parameters 23
3.9. Parameters 23
3.10. Parameters 24
3.11. Parameters 24
3.12. Parameters 24
3.13. Parameters 24
3.14. Parameters 25
3.15. Parameters 25
3.16. Parameters 25
3.17. Parameters 25
3.18. Parameters 26
3.19. Parameters 26
3.20. Parameters 26
3.21. Parameters 27
3.22. Parameters 27
3.23. Parameters 27
3.24. Parameters 27
3.25. Parameters 28
3.26. Parameters 28
3.27. Parameters 28
3.28. Parameters 29
3.29. Parameters 29
3.30. Parameters 29
3.31. Parameters 30
3.32. Parameters 30
3.33. Parameters 30
3.34. Parameters 30
3.35. Parameters 31
3.36. Parameters 31
3.37. Parameters 31
3.38. Parameters 32
3.39. Parameters 32
3.40. Parameters 32
3.41. Parameters 33
3.42. Parameters 33
3.43. Parameters 33
3.44. Parameters 34
3.45. Parameters 34
3.46. Parameters 35
3.47. Parameters 35
3.48. Parameters 35
3.49. Parameters 36
3.50. Parameters 36
3.51. Parameters 36

3.52. Parameters 37
3.53. Parameters 37
3.54. Parameters 37
3.55. Parameters 38
3.56. Parameters 38
3.57. Parameters 38
3.58. Parameters 39
3.59. Parameters 39
3.60. Parameters 40
3.61. Parameters 40
3.62. Parameters 41
3.63. Parameters 41
3.64. Parameters 41
3.65. Parameters 47
3.66. Parameters 47
3.67. Parameters 47
3.68. Parameters 47
3.69. Parameters 48
3.70. Parameters 48
3.71. Parameters 48
3.72. Parameters 48
3.73. Parameters 49
3.74. Parameters 49
3.75. Parameters 49
3.76. Parameters 50
3.77. Parameters 50
3.78. Parameters 50
3.79. Parameters 50
3.80. Parameters 51
3.81. Parameters 51
3.82. Parameters 51
3.83. Parameters 51
3.84. Parameters 52
3.85. Parameters 59
3.86. Parameters 59
3.87. Parameters 60
3.88. Parameters 60
3.89. Parameters 61
3.90. Parameters 61

Chapter 1. Namespace Documentation

Com

Namespaces

- struct Com::Objsys

Com::Objsys

Namespaces

- struct Com::Objsys::Csta

Com::Objsys::Asn1::Runtime

Com::Objsys::Csta

Namespaces

- struct Com::Objsys::Csta::Devices
- struct Com::Objsys::Csta::Phase2

Com::Objsys::Csta::Devices

Classes

- struct Com::Objsys::Csta::Devices::Alcatel4400
- struct Com::Objsys::Csta::Devices::AlcatelOXE
- struct Com::Objsys::Csta::Devices::SamsungSCM
- struct Com::Objsys::Csta::Devices::SiemensHipath3000p2

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.

Definition at line 38 of file Alcatel4400.cs

The Documentation for this struct was generated from the following file:

- Alcatel4400.cs

Com::Objsys::Csta::Phase2

Classes

- struct Com::Objsys::Csta::Phase2::Constants
- struct Com::Objsys::Csta::Phase2::CSTAContext
- struct Com::Objsys::Csta::Phase2::CSTAEncDec
- struct Com::Objsys::Csta::Phase2::CSTAResponseInfo
- struct Com::Objsys::Csta::Phase2::GenericCSTAp2
- struct Com::Objsys::Csta::Phase2::IETF_CSTAp2
- struct Com::Objsys::Csta::Phase2::LicenseException
- struct Com::Objsys::Csta::Phase2::LicenseHelper
- struct Com::Objsys::Csta::Phase2::LicenseOptions
- struct Com::Objsys::Csta::Phase2::PBXSession
- struct Com::Objsys::Csta::Phase2::PBXSessionException
- struct Com::Objsys::Csta::Phase2::PBXSessionHelper
- struct Com::Objsys::Csta::Phase2::PBXSessionHelperPhase2
- struct Com::Objsys::Csta::Phase2::Phase2Opcodes
- struct Com::Objsys::Csta::Phase2::ROSEParseInfo
- struct Com::Objsys::Csta::Phase2::SocketState

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.

Definition at line 46 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

System

System::Collections::Generic

System::IO

System::Linq

System::Net

System::Net::Sockets

System::Reflection

System::Runtime::InteropServices

System::Text

System::Threading

System::Threading::Tasks

Chapter 2. CSTABERPhase2

CSTABERPhase2 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.Phase2`

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 then 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), and its license file has a file type of .lic, 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. If your license file is an osyslic.txt file, these steps are not necessary.

Chapter 3. Class Documentation

Com::Objsys::Csta::Devices::Alcatel4400 class Reference

- Alcatel4400 (string pbxSystem, int port)
- Alcatel4400 (PBXSession sessionObject)
- override CSTAResponseInfo MakeACSEAssociation ( )
- override int EncodeACSEConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer)
- override int EncodeMakeCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice)

Detailed Description

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

Definition at line 44 of file Alcatel4400.cs

The Documentation for this struct was generated from the following file:

- Alcatel4400.cs

Alcatel4400 (string pbxSystem, int port)

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

Table 3.1. Parameters

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

Alcatel4400 (PBXSession sessionObject)

Constructs an instance associated with the given PBXSession object.

Table 3.2. Parameters

sessionObject	A PBXSession object.

override CSTAResponseInfo MakeACSEAssociation ()

Establish an ACSE association with the PBX.

Returns: . A CSTAResponseInfo object.

override int EncodeACSEConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer)

Encodes an ACSE Association Request message.

Table 3.3. 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.

override int EncodeMakeCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice)

Encodes a MakeCall message.

Table 3.4. 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.

Com::Objsys::Csta::Devices::AlcatelOXE class Reference

- AlcatelOXE (string pbxSystem, int port)
- AlcatelOXE (PBXSession sessionObject)

Detailed Description

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

Definition at line 42 of file AlcatelOXE.cs

The Documentation for this struct was generated from the following file:

- AlcatelOXE.cs

AlcatelOXE (string pbxSystem, int port)

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

Table 3.5. Parameters

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

AlcatelOXE (PBXSession sessionObject)

Constructs an instance associated with the given PBXSession object.

Table 3.6. Parameters

sessionObject	A PBXSession object.

ApplicationException class Reference

Com::Objsys::Csta::Phase2::Constants class Reference

- enum ACSEMessageTypes {
  MakeAssociation,
  ReleaseAssociation
  }
- enum CallbackInvocationMechanisms {
  InvokeCallbackThenPostNextRead,
  PostNextReadThenInvokeCallback
  }
- enum CommunicationTypes {
  RawBER,
  IETFBER,
  SiemensBER,
  RawXML,
  SIPXML
  }
- enum Encoding {
  BER,
  XML
  }

- enum PBXModels {
  Unknown,
  GenericBER,
  GenericIETF,
  Panasonic,
  Alcatel4400,
  AlcatelOXO,
  AlcatelOXE,
  SiemensHicom300,
  SiemensHipath3000,
  SiemensCap,
  TadiranCoral,
  SiemensRealitis,
  SiemensHipath4000,
  UnifyOpenscapeX5,
  PhilipsSopho,
  GenericXML,
  GenericSIP,
  UnifyOpenscapeVoice,
  VodiaSNOMOne,
  UnifyOpenscape4000BER,
  SamsungSCM,
  PanasonicNS
  }
- enum XMLSessionMessageTypes {
  StartSession,
  StopSession,
  ResetSession
  }

- const ushort ASN_K_MAXSUBIDS
- const string INVALID_COMM_TYPE
- const string INVALID_PHASE
- const string INVALID_XML_EDITION
- const ushort MAX_NUM_SIZE
- const ushort MAX_RECV_TIMEOUT
- const ushort MAX_SEQUENCE_NUMBER
- const string NO_RESPONSE_FROM_PBX
- const string PBX_UNIVERSAL_FAILURE
- const string ROSE_ENCODE_FAILURE

Public Attributes

- const long MAX_LOGFILE_SIZE

Detailed Description

The Constants class contains some helpful constant and enum definitions.

Definition at line 51 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

enum ACSEMessageTypes

Provides symbolic names for the ACSE message types.

Enumerator:

MakeAssociation

ReleaseAssociation

Definition at line 175 of file Constants.cs

```
{
MakeAssociation,
ReleaseAssociation,
}ACSEMessageTypes;
```

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.

Enumerator:

InvokeCallbackThenPostNextRead

PostNextReadThenInvokeCallback

Definition at line 215 of file Constants.cs

```
{
InvokeCallbackThenPostNextRead,
PostNextReadThenInvokeCallback,
}CallbackInvocationMechanisms;
```

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.

Enumerator:

RawBER

IETFBER

SiemensBER

RawXML

SIPXML

Definition at line 153 of file Constants.cs

```
{
RawBER,
IETFBER,
SiemensBER,
RawXML,
SIPXML,
}CommunicationTypes;
```

enum Encoding

Provides symbolic names for the mechanisms for encoding CSTA messages.

Enumerator:

BER

XML

Definition at line 166 of file Constants.cs

```
{
BER,
XML,
}Encoding;
```

enum PBXModels

Provides symbolic names for different PBX models.

Enumerator:

Unknown

GenericBER

GenericIETF

Panasonic

Alcatel4400

AlcatelOXO

AlcatelOXE

SiemensHicom300

SiemensHipath3000

SiemensCap

TadiranCoral

SiemensRealitis

SiemensHipath4000

UnifyOpenscapeX5

PhilipsSopho

GenericXML

GenericSIP

UnifyOpenscapeVoice

VodiaSNOMOne

UnifyOpenscape4000BER

SamsungSCM

PanasonicNS

Definition at line 122 of file Constants.cs

```
{
Unknown,
GenericBER,
GenericIETF,
Panasonic,
Alcatel4400,
AlcatelOXO,
AlcatelOXE,
SiemensHicom300,
SiemensHipath3000,
SiemensCap,
TadiranCoral,
SiemensRealitis,
SiemensHipath4000,
UnifyOpenscapeX5,
PhilipsSopho,
GenericXML,
GenericSIP,
UnifyOpenscapeVoice,
VodiaSNOMOne,
UnifyOpenscape4000BER,
SamsungSCM,
PanasonicNS,
}PBXModels;
```

enum XMLSessionMessageTypes

Provides symbolic names for the XML session management message types.

Enumerator:

StartSession

StopSession

ResetSession

Definition at line 185 of file Constants.cs

```
{
StartSession,
StopSession,
ResetSession,
}XMLSessionMessageTypes;
```

const ushort ASN_K_MAXSUBIDS

Defines the maximum number of sub-ids that an object id can have.

Definition at line 55 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const string INVALID_COMM_TYPE

Defines a common message for detection of an invalid communication type (should never happen).

Definition at line 91 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const string INVALID_PHASE

Defines a common message for detection of an invalid phase (should never happen).

Definition at line 97 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const string INVALID_XML_EDITION

Defines a common message for detection of an invalid XML edition (should never happen).

Definition at line 103 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const ushort MAX_NUM_SIZE

Defines the maximum number of characters that a text representation of a number can have.

Definition at line 61 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const ushort MAX_RECV_TIMEOUT

Defines the maximum amount of time, in milliseconds, to wait for a response to come in from a PBX.

Definition at line 67 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const ushort MAX_SEQUENCE_NUMBER

Defines the maximum value that the sequence number portion of the invoke id can be.

Definition at line 79 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const string NO_RESPONSE_FROM_PBX

Defines a common message for no response received from the PBX.

Definition at line 108 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const string PBX_UNIVERSAL_FAILURE

Defines a common message for an error returned from the PBX.

Definition at line 114 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

const string ROSE_ENCODE_FAILURE

Defines a common message prefix for a failure to encode the ROSE header.

Definition at line 85 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

Member Data Documentation

const long MAX_LOGFILE_SIZE

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

Definition at line 73 of file Constants.cs

The Documentation for this struct was generated from the following file:

- Constants.cs

Com::Objsys::Csta::Phase2::CSTAContext class Reference

- static Dictionary< string, CSTAContext > cdrCalls
- static object cdrLockObject
- static object contextByThreadIDLO
- static object contextByThreadNumberLO
- static Dictionary< int, CSTAContext > contextsByThreadID
- static Dictionary< ushort, CSTAContext > contextsByThreadNumber
- static Dictionary< string, string > devicesByXref
- static Dictionary< string, CSTAContext > getSFDCalls
- static object getSFDLockObject
- static Dictionary< string, CSTAContext > kmeOperations
- static Dictionary< string, CSTAContext > monitorCalls
- static object monitorLockObject
- static Dictionary< string, List< string > > monitorsByDevice
- static ushort nextThreadNum
- static object panasonicKmeLockObject
- static Dictionary< string, CSTAContext > snapshotCallCalls
- static object snapshotCallLockObject
- static Dictionary< string, CSTAContext > snapshotDevCalls
- static object snapshotDevLockObject
- static bool threadNumberWrapped

-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

Private Attributes

- ushort lastSequenceNum
- byte [] responseFromPBX
- List< byte[]> responsesFromPBX
- int threadID
- ushort threadNumber
- Thread threadObject

- EventWaitHandle waitHandle
- EventWaitHandle xmlResetSessionWaitHandle
- string xmlResponseFromPBX
- List< string > xmlResponsesFromPBX
- EventWaitHandle xmlStopSessionWaitHandle
- CSTAContext ( )
- ~CSTAContext ( )
- static CSTAContext Instance ( )

Detailed Description

The CSTAContext class contains information needed to manage the interaction between the thread and the PBX.

Definition at line 53 of file CSTAContext.cs

The Documentation for this struct was generated from the following file:

- CSTAContext.cs

CSTAContext ()

Default constructor.

~CSTAContext ()

Destructor. Removes this instance from the context list upon garbage collection.

static CSTAContext Instance ()

This method will either return an already created context object or will create a new one.

Returns: . A CSTAContext instance.

Com::Objsys::Csta::Phase2::CSTAEncDec class Reference

- const int DECRYPTION
- const int ENCRYPTION
- uint [] k

- static readonly uint [] bytebit
- static readonly byte [] pc1
- static readonly byte [] pc2
- static uint [][] spbox
- static readonly byte [] totrot

- CSTAEncDec (byte [] key, int dir)
- virtual void ProcessBlock (byte [] inBlock, byte [] outBlock, int offset)

- uint ByteReverse (uint value)
- uint RotateLeft (uint x, uint y)
- uint RotateRight (uint x, uint y)
- void ToBytes (uint ivalue, byte [] b, int offset)
- uint ToInt (byte [] b, int offset)

Com::Objsys::Csta::Phase2::CSTAResponseInfo class Reference

Private Attributes

- byte [] responseFromPBX
- List< byte[]> responsesFromPBX
- int statusCode
- string statusMessage
- string xmlResponseFromPBX
- List< string > xmlResponsesFromPBX

-
-
-
-
-

-
-
-

Detailed Description

Contains information about a PBX operation that was attempted.

Definition at line 51 of file CSTAResponseInfo.cs

The Documentation for this struct was generated from the following file:

- CSTAResponseInfo.cs

Com::Objsys::Csta::Phase2::GenericCSTAp2 class Reference

Protected Attributes

- PBXSession sessionObject
- CSTAContext threadContext

-
-

- virtual CSTAResponseInfo AnswerCall (string deviceToLift)
- virtual CSTAResponseInfo AnswerCall (ConnectionID callToAnswer)
- virtual CSTAResponseInfo AnswerCall (ConnectionID callToAnswer, string deviceToLift)
- virtual CSTAResponseInfo ClearConnection (ConnectionID connToClear)
- virtual CSTAResponseInfo ClearDoNotDisturb (string targetDevice)
- virtual CSTAResponseInfo ClearMessageWaiting (string targetDevice)
- virtual CSTAResponseInfo ConferenceCall (ConnectionID heldCall, ConnectionID activeCall)
- virtual CSTAResponseInfo ConsultationCall (ConnectionID existingCall, string targetDevice)
- virtual CSTAResponseInfo DivertCall (string divertFrom, string divertTo)
- int EncodeROSERequestHeader (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, Phase2Opcodes.Opcodes opcode)
- GenericCSTAp2 (string pbxSystem, int port)
- GenericCSTAp2 (PBXSession sessionObject)

- virtual CSTAResponseInfo HoldCall (ConnectionID callToHold)
- virtual CSTAResponseInfo MakeACSEAssociation ( )
- virtual CSTAResponseInfo MakeCall (string callingDevice, string calledDevice)
- virtual CSTAResponseInfo MonitorStart (string deviceToMonitor)
- virtual CSTAResponseInfo MonitorStart (ConnectionID callToMonitor)
- virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID crossRefID)
- virtual CSTAResponseInfo MonitorStop (string monitoredDevice)
- virtual CSTAResponseInfo MonitorStop (long monitoredDevice)
- virtual CSTAResponseInfo QueryDevice (string deviceToQuery)
- virtual CSTAResponseInfo ReleaseACSEAssociation ( )
- virtual CSTAResponseInfo RetrieveCall (ConnectionID callToRetrieve)
- virtual CSTAResponseInfo SetAgentState (string targetDevice, AgentParameter agentParam)
- virtual CSTAResponseInfo SetDoNotDisturb (string targetDevice)
- virtual CSTAResponseInfo SetMessageWaiting (string targetDevice)
- virtual CSTAResponseInfo SetOrClearForwarding (string fromDevice, ForwardingType fwdType, string toDevice)
- virtual CSTAResponseInfo SnapshotDevice (string deviceToSnapshot)
- virtual CSTAResponseInfo TransferCall (ConnectionID initiatedCall, ConnectionID originalCall)
- virtual CSTAResponseInfo TransferCall (string heldDevice, string connectedDevice)

- virtual int EncodeACSEConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer)
- virtual int EncodeACSEReleaseRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer)
- virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToLift)
- virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer)
- virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer, string deviceToLift)
- virtual int EncodeClearConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID connToClear)
- virtual int EncodeConferenceCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID heldCall, ConnectionID activeCall)
- virtual int EncodeConsultationCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID existingCall, string targetDevice)

- virtual int EncodeDivertCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string divertFrom, string divertTo)
- virtual int EncodeHoldCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToHold)
- virtual int EncodeMakeCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice)
- virtual int EncodeMonitorStartRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToMonitor)
- virtual int EncodeMonitorStartRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToMonitor)
- virtual int EncodeMonitorStopRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, MonitorCrossRefID crossRefObj)
- virtual int EncodeQueryDeviceRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToQuery)
- virtual int EncodeRetrieveCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToRetrieve)
- virtual int EncodeSetAgentStateRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, AgentParameter agentParam)
- virtual int EncodeSetOrClearDNDRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool dndOn)
- virtual int EncodeSetOrClearFwdRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string fromDevice, ForwardingType fwdType, string toDevice)
- virtual int EncodeSetOrClearMWRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool indicatorOn)
- virtual int EncodeSnapshotDeviceRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToSnapshot)
- virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID initiatedCall, ConnectionID originalCall)
- virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string heldDevice, string connectedDevice)
- CSTA_ROSE_PDU InterpretROSEResponse (CSTAResponseInfo response)
- void InterpretACSEResponse (CSTAResponseInfo response)

Detailed Description

Implements CSTA phase 2 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.

Definition at line 46 of file GenericCSTAp2.cs

The Documentation for this struct was generated from the following file:

- GenericCSTAp2.cs

virtual CSTAResponseInfo AnswerCall (string deviceToLift)

Answers a call.

Table 3.7. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo AnswerCall (ConnectionID callToAnswer)

Answers a call.

Table 3.8. Parameters

callToAnswer	The ConnectionID of the call to answer.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo AnswerCall (ConnectionID callToAnswer, string deviceToLift)

Answers a call.

Table 3.9. Parameters

callToAnswer	The connection id of the call to answer.
deviceToLift	The device (e.g., extension number) that is to answer the call.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo ClearConnection (ConnectionID connToClear)

Clears a connection.

Table 3.10. Parameters

connToClear	The ConnectionID of the connection to clear.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo ClearDoNotDisturb (string targetDevice)

Turns off the Do Not Disturb functionality for a phone.

Table 3.11. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo ClearMessageWaiting (string targetDevice)

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

Table 3.12. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo ConferenceCall (ConnectionID heldCall, ConnectionID activeCall)

Brings a held call into conference with an active call.

Table 3.13. Parameters

heldCall	The held call to be brought into conference.
activeCall	The active call.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo ConsultationCall (ConnectionID existingCall, string targetDevice)

Instruct the PBX to do a consultation call.

Table 3.14. 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.

virtual CSTAResponseInfo DivertCall (string divertFrom, string divertTo)

Diverts a call from a source to a destination.

Table 3.15. Parameters

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

Returns: . A CSTAResponseInfo object.

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

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

Table 3.16. 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.

GenericCSTAp2 (string pbxSystem, int port)

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

Table 3.17. Parameters

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

port	Port on which the PBX listens for CSTA messages.

GenericCSTAp2 (PBXSession sessionObject)

Constructs an instance associated with the given PBXSession object.

Table 3.18. Parameters

sessionObject	A PBXSession object.

virtual CSTAResponseInfo HoldCall (ConnectionID callToHold)

Instruct the PBX to hold a call.

Table 3.19. Parameters

callToHold	The ConnectionID of the call to be held.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo MakeACSEAssociation ()

Establish an ACSE association with the PBX.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo MakeCall (string callingDevice, string calledDevice)

Instruct the PBX to place a call.

Table 3.20. 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.

virtual CSTAResponseInfo MonitorStart (string deviceToMonitor)

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

Table 3.21. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo MonitorStart (ConnectionID callToMonitor)

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

Table 3.22. Parameters

callToMonitor	The call to monitor.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo MonitorStop (MonitorCrossRefID crossRefID)

Stop a previously started PBX monitor request.

Table 3.23. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo MonitorStop (string monitoredDevice)

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.

Table 3.24. 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.

virtual CSTAResponseInfo MonitorStop (long monitored-Device)

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.

Table 3.25. 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.

virtual CSTAResponseInfo QueryDevice (string device-ToQuery)

Queries a device.

Table 3.26. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo ReleaseACSEAssociation ()

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.

virtual CSTAResponseInfo RetrieveCall (ConnectionID callToRetrieve)

Retrieves a held call.

Table 3.27. Parameters

callToRetrieve	The ConnectionID of the call to retrieve.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo SetAgentState (string targetDevice, AgentParameter agentParam)

Sets the state of an agent.

Table 3.28. Parameters

targetDevice	The device for which the agent state is to be set.
agentParam	An AgentParameter object.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo SetDoNotDisturb (string targetDevice)

Sets the Do Not Disturb feature for a phone.

Table 3.29. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo SetMessageWaiting (string targetDevice)

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

Table 3.30. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo SetOrClearForwarding (string fromDevice, ForwardingType fwdType, string toDevice)

Sets or clears forwarding for a device.

Table 3.31. Parameters

fromDevice	The device for which forwarding is to be turned on or off.
fwdType	A ForwardingType object. This object specifies the type of forwarding and whether it is to be turned on or off.
toDevice	If fwdType indicates that forwarding is to be turned on, then this argument is the device to which calls are to be forwarded. Otherwise, this argument is ignored and can be null.

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo SnapshotDevice (string deviceToSnapshot)

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

Table 3.32. Parameters

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

Returns: . A CSTAResponseInfo object.

virtual CSTAResponseInfo TransferCall (ConnectionID initiatedCall, ConnectionID originalCall)

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

Table 3.33. 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.

virtual CSTAResponseInfo TransferCall (string heldDevice, string connectedDevice)

Transfers a call from one device to another.

Table 3.34. 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.

virtual int EncodeACSEConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer)

Encodes an ACSE Association Request message.

Table 3.35. 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.

virtual int EncodeACSEReleaseRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer)

Encodes an ACSE Release Request message.

Table 3.36. 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.

virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToLift)

Encodes an AnswerCall message.

Table 3.37. 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.

virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer)

Encodes an AnswerCall message.

Table 3.38. 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.

virtual int EncodeAnswerCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToAnswer, string deviceToLift)

Encodes an AnswerCall message.

Table 3.39. 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.

virtual int EncodeClearConnectionRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID connToClear)

Encodes a ClearConnection message.

Table 3.40. 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.

virtual int EncodeConferenceCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID heldCall, ConnectionID activeCall)

Encodes a ConferenceCall message.

Table 3.41. 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.

virtual int EncodeConsultationCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID existingCall, string targetDevice)

Encodes a ConsultationCall message.

Table 3.42. 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.

virtual int EncodeDivertCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string divertFrom, string divertTo)

Encodes a DivertCall message.

Table 3.43. Parameters

response	A CSTA ResponseInfo object.

encodeBuffer	An encode buffer object into which the message will be encoded.
divertFrom	Identifier (e.g., phone number) of the device from which the call is to be diverted.
divertTo	Identifier (e.g., phone number) of the device to which the call is to be diverted.

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

virtual int EncodeHoldCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToHold)

Encodes a HoldCall message.

Table 3.44. 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.

virtual int EncodeMakeCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string callingDevice, string calledDevice)

Encodes a MakeCall message.

Table 3.45. 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.

virtual int EncodeMonitorStartRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToMonitor)

Encodes a MonitorStart message to monitor a device.

Table 3.46. 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.

virtual int EncodeMonitorStartRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToMonitor)

Encodes a MonitorStart message to monitor a call.

Table 3.47. 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.

virtual int EncodeMonitorStopRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, MonitorCrossRefID crossRefObj)

Encodes a MonitorStop message.

Table 3.48. 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.

virtual int EncodeQueryDeviceRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToQuery)

Encodes a QueryDevice message.

Table 3.49. Parameters

response	A CSTA ResponseInfo object.
encodeBuffer	An encode buffer object into which the message will be encoded.
deviceToQuery	The identification (e.g., phone number) of the device to query.

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

virtual int EncodeRetrieveCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID callToRetrieve)

Encodes a RetrieveCall message.

Table 3.50. 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.

virtual int EncodeSetAgentStateRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, AgentParameter agentParam)

Encodes a SetFeature message to set an agent state.

Table 3.51. Parameters

response	A CSTA ResponseInfo object.
encodeBuffer	An encode buffer object into which the message will be encoded.
targetDevice	The device for which the agent state will be set.
agentParam	An AgentParameter object.

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

virtual int EncodeSetOrClearDNDRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool dndOn)

Encodes a SetFeature message to set or clear the Do Not Disturb setting.

Table 3.52. 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.

virtual int EncodeSetOrClearFwdRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string fromDevice, ForwardingType fwdType, string toDevice)

Encodes a SetFeature message to set or clear forwarding.

Table 3.53. Parameters

response	A CSTA ResponseInfo object.
encodeBuffer	An encode buffer object into which the message will be encoded.
fromDevice	The device for which forwarding is to be turned on or off.
fwdType	A ForwardingType object. This object specifies the type of forwarding and whether it is to be turned on or off.
toDevice	If fwdType indicates that forwarding is to be turned on, then this argument is the device to which calls are to be forwarded. Otherwise, this argument is ignored and can be null.

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

virtual int EncodeSetOrClearMWRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string targetDevice, bool indicatorOn)

Encodes a SetFeature message to set or clear the Message Waiting indicator.

Table 3.54. 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.

virtual int EncodeSnapshotDeviceRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string deviceToSnapshot)

Encodes a SnapshotDevice message.

Table 3.55. 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.

virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, ConnectionID initiatedCall, ConnectionID originalCall)

Encodes a TransferCall message.

Table 3.56. 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.

virtual int EncodeTransferCallRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, string heldDevice, string connectedDevice)

Encodes a TransferCall message.

Table 3.57. Parameters

response	A CSTA ResponseInfo object.

encodeBuffer	An encode buffer object into which the message will be encoded.
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: . The length of the encoded message, or -1 if an error occurred.

Com::Objsys::Csta::Phase2::IETF_CSTAp2 class Reference

- IETF_CSTAp2 (string pbxSystem, int port)
- IETF_CSTAp2 (PBXSession sessionObject)

Detailed Description

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

Definition at line 43 of file IETF_CSTAp2.cs

The Documentation for this struct was generated from the following file:

- IETF_CSTAp2.cs

IETF_CSTAp2 (string pbxSystem, int port)

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

Table 3.58. Parameters

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

IETF_CSTAp2 (PBXSession sessionObject)

Constructs an instance associated with the given PBXSession object.

Table 3.59. Parameters

sessionObject	A PBXSession object.

Com::Objsys::Csta::Phase2::LicenseException class Reference

- LicenseException (string message)

Detailed Description

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

Definition at line 52 of file LicenseException.cs

The Documentation for this struct was generated from the following file:

- LicenseException.cs

Com::Objsys::Csta::Phase2::LicenseHelper class Reference

- static object licenseLO
- static ushort sessionCount

- static bool CheckLicenseSettings (LicenseOptions.BERPhases phase)
- static bool CheckLicenseSettings (LicenseOptions.XMLEditions edition)
- static void FreeLicense (bool close)
- static string GenNotEnabledMsg (Constants.Encoding encoding, ushort phaseOrEdition)
- static void HandleException (PBXSession sessionObject, string text)

static bool CheckLicenseSettings (LicenseOptions.BERPhases phase)

Returns true if the indicated BER phase is enabled in the license, false otherwise.

Table 3.60. Parameters

phase	Indicates the BER phase number.

Returns: . True if the capability is enabled in the license, false otherwise.

static bool CheckLicenseSettings (LicenseOptions.XMLEditions edition)

Returns true if the indicated XML edition is enabled in the license, false otherwise.

Table 3.61. Parameters

edition	Indicates the XML edition number.

Returns: . True if the capability is enabled in the license, false otherwise.

static void FreeLicense (bool close)

This method checks in a license.

Table 3.62. Parameters

close	Indicates whether to close the RLM handle if RLM is being used.

static string GenNotEnabledMsg (Constants.Encoding encoding, ushort phaseOrEdition)

Returns a string indicating that a particular capability is not enabled in the user's CSTADLL license.

Table 3.63. Parameters

encoding	Indicates BER or XML encoding.
phaseOrEdition	Indicates the phase number (for BER) or the edition number (for XML)

Returns: . The "not enabled" message as a string.

static void HandleException (PBXSession sessionObject, string text)

This method either throws an exception or invokes an exception callback.

Table 3.64. Parameters

sessionObject	The PBXSession object.
text	The text of the exception message.

Com::Objsys::Csta::Phase2::LicenseOptions class Reference

- enum BERPhases {
  BERPhase1,
  BERPhase2,
  BERPhase3
  }
- enum XMLEditions {
  XMLEdition3,

XMLEdition4,
XMLEdition5,
XMLEdition6
}

Private Attributes

- bool [] enabledBERPhases
- bool [] enabledXMLEditions

-

-

Detailed Description

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

Definition at line 53 of file LicenseOptions.cs

The Documentation for this struct was generated from the following file:

- LicenseOptions.cs

enum BERPhases

Values indices into the array of booleans that defines what BER phases are enabled in the license.

Enumerator:

BERPhase1

BERPhase2

BERPhase3

Definition at line 59 of file LicenseOptions.cs

```
{
BERPhase1,
BERPhase2,
BERPhase3,
}BERPhases;
```

enum XMLEditions

Values indices into the array of booleans that defines what XML editions are enabled in the license.

Enumerator:

XMLEdition3

XMLEdition4

XMLEdition5

XMLEdition6

Definition at line 82 of file LicenseOptions.cs

```
{
XMLEdition3,
XMLEdition4,
XMLEdition5,
XMLEdition6,
}XMLEditions;
```

Com::Objsys::Csta::Phase2::PBXSession class Reference

Private Attributes

- bool asyncReadInProgress
- Constants.CallbackInvocationMechanisms callbackInvocationMechanism
- AsyncCallback cdrCallback
- AsyncCallback clientCallback
- Constants.CommunicationTypes commType
- bool connected
- ConnectionCallback connectionLostCallback
- bool debugMode
- string deviceType
- bool discardOldResponses
- AsyncExceptionCallback exceptionCallback
- long lastInvokeId
- List< long > lateInvokeIds
- int maxReceiveTimeout
- Constants.Encoding messageEncoding
- Constants.PBXModels pbxModel
- Socket pbxSocket
- string pbxSystem

- ushort phase
- int port
- AsyncCallback systemStatusCallback
- string uaSIPContact
- XMLAsyncCallback xmlCDRCallback
- XMLAsyncCallback xmlClientCallback
- ushort xmlEdition
- bool xmlImmediateSystemStatus
- string xmlLastInvokeId
- List< string > xmlLateInvokeIds
- CSTAContext xmlResetSessionContext
- EventWaitHandle xmlResetSessionTerminate
- string xmlSessionID
- CSTAContext xmlStopSessionContext
- XMLAsyncCallback xmlSystemStatusCallback

- static Dictionary< IntPtr, PBXSession > sessionList
- static object sessionLockObject

- 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 (byte [] message, int messageLength, CSTAContext threadContext)
- void SendMessage (string messageType, byte [] message, int messageLength, CSTAContext threadContext)
- void SendXMLMessage (string strMessage, CSTAContext threadContext)
- void SendXMLMessage (string messageType, 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)
- static void Init ( )
- void AlcatelInit (CSTAContext threadContext)
- void Connect (CSTAContext threadContext)
- void SendMessageInternal (byte [] message, int messageLength, CSTAContext threadContext)
- void WaitForResetSessionResponse (CSTAContext threadContext)
- void WaitForStopSessionResponse (CSTAContext threadContext)
- void WaitForResponse (CSTAContext threadContext, EventWaitHandle waitHandle)

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.

Definition at line 73 of file PBXSession.cs

The Documentation for this struct was generated from the following file:

- PBXSession.cs

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.

Table 3.65. Parameters

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

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.

Table 3.66. 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.

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.

Table 3.67. Parameters

threadContext	The context object for the caling thread.

delegate void ConnectionCallback (PBXSession sessionObject)

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

Table 3.68. Parameters

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

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.

Table 3.69. Parameters

threadContext	The thread context object.

PBXSession (string pbxSystem, int port)

Constructs a PBXSession object.

Table 3.70. Parameters

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

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.

Table 3.71. 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.

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

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

Table 3.72. Parameters

message	Byte array containing the encoded message to send.
messageLength	The length of the encoded message.

threadContext	The thread context object.

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

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

Table 3.73. 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.

void SendXMLMessage (string strMessage, CSTAContext threadContext)

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

Table 3.74. Parameters

strMessage	The XML message to send.
threadContext	The thread context object.

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

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

Table 3.75. 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.

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

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

Table 3.76. 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.

void WaitForROSEResponse (CSTAContext threadContext)

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

Table 3.77. Parameters

threadContext	The CSTAContext object associated with the calling thread.

void WaitForXMLResponse (CSTAContext threadContext)

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

Table 3.78. Parameters

threadContext	The CSTAContext object associated with the calling thread.

delegate void XMLAsyncCallback (PBXSession sessionObject, string message)

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

Table 3.79. 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.

static void Init ()

Performs license initialization from the generated objects. This method gets called instead of SetKey() or SetKey2(). This is here in case a user just wants to use the generated classes and not the helper classes.

void AlcatelInit (CSTAContext threadContext)

Sends the initialization byte to an Alcatel PBX and receives the response.

Table 3.80. Parameters

threadContext	A context object.

void Connect (CSTAContext threadContext)

This method establishes TCP/IP connectivity to a PBX without starting an asynchronous read to receive messages sent by the PBX. If the PBX is an XML PBX or UA that immediately sends a System Status message once TCP/IP connectivity is established, this message will be received, and a response will be sent.

Table 3.81. Parameters

threadContext	A CSTAContext object.

void SendMessageInternal (byte[] message, int messageLength, CSTAContext threadContext)

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

Table 3.82. Parameters

message	Byte array containing the encoded message to send.
messageLength	The length of the encoded message.
threadContext	The thread context object.

void WaitForResetSessionResponse (CSTAContext threadContext)

This method waits for a response to a ResetSession message.

Table 3.83. Parameters

threadContext	The CSTAContext object associated with the calling thread.

void WaitForStopSessionResponse (CSTAContext threadContext)

This method waits for a response to a StopSession message.

Table 3.84. Parameters

threadContext	The CSTAContext object associated with the calling thread.

Com::Objsys::Csta::Phase2::PBXSessionException class Reference

- PBXSessionException (string message)

Detailed Description

Defines an exception that occurs while communicating with a PBX.

Definition at line 51 of file PBXSessionException.cs

The Documentation for this struct was generated from the following file:

- PBXSessionException.cs

Com::Objsys::Csta::Phase2::PBXSessionHelper class Reference

-
-
-
-
-

- static List< string > routeMessageTags
- static object routeMessageTagsLO

- static void HandleException (PBXSession sessionObject, string text)
- static void HandleSocketException (SocketException se, PBXSession sessionObject, Socket pbxSocket, ushort location)
- static void QueueNextRead (bool newSocketState, SocketState ss, PBXSession sessionObject)
- static void Read_Callback (IAsyncResult ar)

- static void HandleBER (PBXSession sessionObject, SocketState ss, Socket pbxSocket, IAsyncResult ar)

Detailed Description

This class holds static properties that affect all PBX sessions.

Definition at line 56 of file PBXSessionHelper.cs

The Documentation for this struct was generated from the following file:

- PBXSessionHelper.cs

Com::Objsys::Csta::Phase2::PBXSessionHelperPhase2 class Reference

- static void HandleBERPhase2 (PBXSession sessionObject, SocketState ss, Socket pbxSocket, IAsyncResult ar)
- static int EncodeEscapeResponse (Asn1BerEncodeBuffer encodeBuffer, PBXSession sessionObject, long invokeid)
- static int EncodeSSResponse (Asn1BerEncodeBuffer encodeBuffer, PBXSession sessionObject)
- static ROSEParseInfo ParseROSE (PBXSession sessionObject, CSTA_ROSE_PDU rosePDU)

Detailed Description

This class contains utility methods used by PBXSessionHelper for BER phase 2.

Definition at line 48 of file PBXSessionHelperPhase2.cs

The Documentation for this struct was generated from the following file:

- PBXSessionHelperPhase2.cs

Com::Objsys::Csta::Phase2::Phase2Opcodes class Reference

- enum Opcodes {
  alternateCall_CODE= 1,
  answerCall_CODE,
  callCompletion_CODE,
  clearCall_CODE,
  clearConnection_CODE,
  conferenceCall_CODE,
  consultationCall_CODE,
  divertCall_CODE,
  holdCall_CODE,
  makeCall_CODE,
  makePredictiveCall_CODE,
  queryDevice_CODE,
  reconnectCall_CODE,

retrieveCall_CODE,
setFeature_CODE,
transferCall_CODE,
associateData_CODE,
parkCall_CODE,
sendDTMFTones_CODE,
singleStepConf_CODE,
cSTAEventReport_CODE,
routeRequest_CODE= 31,
reRouteRequest_CODE,
routeSelectRequest_CODE,
routeUsedRequest_CODE,
routeEndRequest_CODE,
singleStepTrans_CODE= 50,
escapeService_CODE,
systemStatus_CODE,
monitorStart_CODE= 71,
changeMonitorFilter_CODE,
monitorStop_CODE,
snapshotDevice_CODE,
snapshotCall_CODE,
startDataPath_CODE= 110,
stopDataPath_CODE,
sendData_CODE,
sendMulticastData_CODE,
sendBroadcastData_CODE,
suspendDataPath_CODE,
dataPathSuspended_CODE,
resumeDataPath_CODE,
dataPathResumed_CODE,
fastData_CODE,
concatenateMessage_CODE= 500,
deleteMessage_CODE,
playMessage_CODE,
queryVoiceAttribute_CODE,
reposition_CODE,
resume_CODE,
review_CODE,
setVoiceAttribute_CODE,
stop_CODE,
suspend_CODE,
synthesizeMessage_CODE,
recordMessage_CODE
}

Detailed Description

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

Definition at line 43 of file Phase2Opcodes.cs

The Documentation for this struct was generated from the following file:

- Phase2Opcodes.cs

enum Opcodes

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

Enumerator:

alternateCall_CODE

answerCall_CODE

callCompletion_CODE

clearCall_CODE

clearConnection_CODE

conferenceCall_CODE

consultationCall_CODE

divertCall_CODE

holdCall_CODE

makeCall_CODE

makePredictiveCall_CODE

queryDevice_CODE

reconnectCall_CODE

retrieveCall_CODE

setFeature_CODE

transferCall_CODE

associateData_CODE

parkCall_CODE

sendDTMFTones_CODE

singleStepConf_CODE

cSTAEventReport_CODE

routeRequest_CODE

reRouteRequest_CODE

routeSelectRequest_CODE

routeUsedRequest_CODE

routeEndRequest_CODE

singleStepTrans_CODE

escapeService_CODE

systemStatus_CODE

monitorStart_CODE

changeMonitorFilter_CODE

monitorStop_CODE

snapshotDevice_CODE

snapshotCall_CODE

startDataPath_CODE

stopDataPath_CODE

sendData_CODE

sendMulticastData_CODE

sendBroadcastData_CODE

suspendDataPath_CODE

dataPathSuspended_CODE

resumeDataPath_CODE

dataPathResumed_CODE

fastData_CODE

concatenateMessage_CODE

deleteMessage_CODE

playMessage_CODE

queryVoiceAttribute_CODE

reposition_CODE

resume_CODE

review_CODE

setVoiceAttribute_CODE

stop_CODE

suspend_CODE

synthesizeMessage_CODE

recordMessage_CODE

Definition at line 49 of file Phase2Opcodes.cs

```
{
alternateCall_CODE= 1,
answerCall_CODE,
callCompletion_CODE,
clearCall_CODE,
clearConnection_CODE,
conferenceCall_CODE,
consultationCall_CODE,
divertCall_CODE,
holdCall_CODE,
makeCall_CODE,
makePredictiveCall_CODE,
queryDevice_CODE,
reconnectCall_CODE,
retrieveCall_CODE,
setFeature_CODE,
transferCall_CODE,
associateData_CODE,
parkCall_CODE,
sendDTMFTones_CODE,
singleStepConf_CODE,
cSTAEventReport_CODE,
routeRequest_CODE= 31,
reRouteRequest_CODE,
routeSelectRequest_CODE,
routeUsedRequest_CODE,
routeEndRequest_CODE,
singleStepTrans_CODE= 50,
escapeService_CODE,
systemStatus_CODE,
monitorStart_CODE= 71,
changeMonitorFilter_CODE,
monitorStop_CODE,
snapshotDevice_CODE,
snapshotCall_CODE,
startDataPath_CODE= 110,
stopDataPath_CODE,
sendData_CODE,
sendMulticastData_CODE,
sendBroadcastData_CODE,
suspendDataPath_CODE,
dataPathSuspended_CODE,
resumeDataPath_CODE,
dataPathResumed_CODE,
fastData_CODE,
concatenateMessage_CODE= 500,
deleteMessage_CODE,
playMessage_CODE,
queryVoiceAttribute_CODE,
reposition_CODE,
resume_CODE,
review_CODE,
setVoiceAttribute_CODE,
stop_CODE,
```

```
suspend_CODE,
synthesizeMessage_CODE,
recordMessage_CODE,
}Opcodes;
```

Com::Objsys::Csta::Phase2::ROSEParseInfo class Reference

-
-
-
-
-
-
-
-
-
-
-

Detailed Description

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

Definition at line 49 of file ROSEParseInfo.cs

The Documentation for this struct was generated from the following file:

- ROSEParseInfo.cs

Com::Objsys::Csta::Devices::SamsungSCM class Reference

Private Attributes

- EventWaitHandle sendLinkAliveTerminate

- virtual CSTAResponseInfo LinkSetup (bool returnNumberPlan)
- sealed override CSTAResponseInfo MakeACSEAssociation ( )

- SamsungSCM (string pbxSystem, int port)
- SamsungSCM (PBXSession sessionObject)
- static int EncodeLinkAliveRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, CSTA-Context threadContext, PBXSession sessionObject)
- static void SendLinkAlive (object arg)
- virtual int EncodeLinkSetupRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, bool returnNumberPlan)

Detailed Description

Implements CSTA phase 2 operations for the Samsung SCM PBX device.

Definition at line 46 of file SamsungSCM.cs

The Documentation for this struct was generated from the following file:

- SamsungSCM.cs

virtual CSTAResponseInfo LinkSetup (bool returnNumberPlan)

Establishes initial communication with the Samsung SCM.

Table 3.85. Parameters

returnNumberPlan	Indicates whether the user number plan is included in the response message.

Returns: . A CSTAResponseInfo object.

sealed override CSTAResponseInfo MakeACSEAssociation ()

Throws an exception since this PBX device doesn't do ACSE associations.

Returns: . Nothing. An exception is thrown.

SamsungSCM (string pbxSystem, int port)

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

Table 3.86. Parameters

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

port	Port on which the PBX listens for CSTA messages.

SamsungSCM (PBXSession sessionObject)

Constructs an instance associated with the given PBXSession object.

Table 3.87. Parameters

sessionObject	A PBXSession object.

virtual int EncodeLinkSetupRequest (CSTAResponseInfo response, Asn1BerEncodeBuffer encodeBuffer, bool returnNumberPlan)

Encodes the Samsung CSTA_LINK_SETUP message.

Table 3.88. Parameters

response	A CSTAResponseInfo object.
encodeBuffer	An encode buffer object into which the message will be encoded.
returnNumberPlan	Indicates whether the user number plan is included in the response message.

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

Com::Objsys::Csta::Devices::SiemensHipath3000p2 class Reference

- SiemensHipath3000p2 (string pbxSystem, int port)
- SiemensHipath3000p2 (PBXSession sessionObject)

Detailed Description

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

Definition at line 43 of file SiemensHipath3000p2.cs

The Documentation for this struct was generated from the following file:

- SiemensHipath3000p2.cs

SiemensHipath3000p2 (string pbxSystem, int port)

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

Table 3.89. Parameters

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

SiemensHipath3000p2 (PBXSession sessionObject)

Constructs an instance associated with the given PBXSession object.

Table 3.90. Parameters

sessionObject	A PBXSession object.

Com::Objsys::Csta::Phase2::SocketState class Reference

Private Attributes

- byte [] ackBuffer
- int ackLength
- int allocatedLength
- uint asn1Tag
- int bytesRequested
- uint classForm
- Socket comSocket
- int currentHeaderBegin
- int fragmentLength
- uint idCode
- bool ietfLengthRequested
- Asn1Choice invokeID
- bool isIetfLengthComplete
- bool isLengthComplete
- bool isLengthStarted
- bool isMessageComplete

- bool isSIPBlockComplete
- bool isTagComplete
- bool isTagStarted
- bool isXMLPrefixComplete
- bool lastSIPCRLF
- byte [] readBuffer
- List< byte[]> readBuffers
- bool secondIetfByteRequested
- int sipContentLength
- string sipMessageType
- uint sipSequenceNumber
- ushort tagReadCount
- CSTAContext threadContext
- ushort threadNumber
- int totalLength
- byte [] xmlHeader

- const ushort ASN_K_MEMBUFINITIAL
- void Reset ( )
- SocketState ( )
- void ValidateBufferLength ( )

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.

Definition at line 59 of file SocketState.cs

The Documentation for this struct was generated from the following file:

- SocketState.cs

const ushort ASN_K_MEMBUFINITIAL

The amount of memory we initially allocate for the receive buffer.

Definition at line 451 of file SocketState.cs

The Documentation for this struct was generated from the following file:

- SocketState.cs

void Reset ()

Resets the object for re-use.

SocketState ()

Default constructor. Creates an EventWaitHandle object.

void ValidateBufferLength ()

Checks to see if the read buffer is large enough to receive the expected fragment. If it isn't, we reallocate.

Chapter 4. File Documentation

_SeqOfFloatLicProductInfo.cs File Reference

Detailed Description

Definition in file _SeqOfFloatLicProductInfo.cs

Alcatel4400.cs File Reference

Classes

- struct Com::Objsys::Csta::Devices::Alcatel4400

Namespaces

- struct Com::Objsys::Asn1::Runtime
- struct Com::Objsys::Csta::Devices
- struct System
- struct System::Collections::Generic
- struct System::IO
- struct System::Text

Detailed Description

Definition in file Alcatel4400.cs

AlcatelOXE.cs File Reference

Classes

- struct Com::Objsys::Csta::Devices::AlcatelOXE

Namespaces

- struct Com::Objsys::Csta::Devices

Detailed Description

Definition in file AlcatelOXE.cs

AlcatelOXO.cs File Reference

Detailed Description

Definition in file AlcatelOXO.cs

Constants.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::Constants

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file Constants.cs

CSTAContext.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::CSTAContext

Namespaces

- struct Com::Objsys::Csta::Phase2
- struct System::Threading

Detailed Description

Definition in file CSTAContext.cs

CSTAEncDec.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::CSTAEncDec

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file CSTAEncDec.cs

CSTAResponseInfo.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::CSTAResponseInfo

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file CSTAResponseInfo.cs

FloatLicInfo.cs File Reference

Detailed Description

Definition in file FloatLicInfo.cs

FloatLicProductInfo.cs File Reference

Detailed Description

Definition in file FloatLicProductInfo.cs

GenericCSTAp2.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::GenericCSTAp2

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file GenericCSTAp2.cs

IETF_CSTAp2.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::IETF_CSTAp2

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file IETF_CSTAp2.cs

LicenseBitFlags.cs File Reference

Detailed Description

Definition in file LicenseBitFlags.cs

LicenseChoice.cs File Reference

Detailed Description

Definition in file LicenseChoice.cs

LicenseChoice_hosts.cs File Reference

Detailed Description

Definition in file LicenseChoice_hosts.cs

LicenseData.cs File Reference

Detailed Description

Definition in file LicenseData.cs

LicenseData_licProcIds.cs File Reference

Detailed Description

Definition in file LicenseData_licProcIds.cs

LicensedProduct.cs File Reference

Detailed Description

Definition in file LicensedProduct.cs

LicenseException.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::LicenseException

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file LicenseException.cs

LicenseHelper.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::LicenseHelper

Namespaces

- struct Com::Objsys::Csta::Phase2
- struct System::Reflection

Detailed Description

Definition in file LicenseHelper.cs

LicenseHost.cs File Reference

Detailed Description

Definition in file LicenseHost.cs

LicenseHost_id.cs File Reference

Detailed Description

Definition in file LicenseHost_id.cs

LicenseOptions.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::LicenseOptions

Namespaces

- struct Com::Objsys::Csta::Phase2
- struct System::Linq
- struct System::Threading::Tasks

Detailed Description

Definition in file LicenseOptions.cs

LicenseUserInfo.cs File Reference

Detailed Description

Definition in file LicenseUserInfo.cs

LicenseValidityPeriod.cs File Reference

Detailed Description

Definition in file LicenseValidityPeriod.cs

PanasonicKXNS.cs File Reference

Detailed Description

Definition in file PanasonicKXNS.cs

PanasonicKXTDA.cs File Reference

Detailed Description

Definition in file PanasonicKXTDA.cs

PanasonicKXTDE.cs File Reference

Detailed Description

Definition in file PanasonicKXTDE.cs

PanasonicNCP.cs File Reference

Detailed Description

Definition in file PanasonicNCP.cs

PanasonicNXS.cs File Reference

Detailed Description

Definition in file PanasonicNXS.cs

PBXSession.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::PBXSession

Namespaces

- struct Com::Objsys::Csta::Phase2
- struct System::Net
- struct System::Net::Sockets

Detailed Description

Definition in file PBXSession.cs

PBXSessionException.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::PBXSessionException

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file PBXSessionException.cs

PBXSessionHelper.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::PBXSessionHelper

Namespaces

- struct Com::Objsys::Csta::Phase2

- struct System::Runtime::InteropServices

Detailed Description

Definition in file PBXSessionHelper.cs

PBXSessionHelperEd3.cs File Reference

Detailed Description

Definition in file PBXSessionHelperEd3.cs

PBXSessionHelperEd4.cs File Reference

Detailed Description

Definition in file PBXSessionHelperEd4.cs

PBXSessionHelperEd5.cs File Reference

Detailed Description

Definition in file PBXSessionHelperEd5.cs

PBXSessionHelperEd6.cs File Reference

Detailed Description

Definition in file PBXSessionHelperEd6.cs

PBXSessionHelperPhase1.cs File Reference

Detailed Description

Definition in file PBXSessionHelperPhase1.cs

PBXSessionHelperPhase2.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::PBXSessionHelperPhase2

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file PBXSessionHelperPhase2.cs

PBXSessionHelperPhase3.cs File Reference

Detailed Description

Definition in file PBXSessionHelperPhase3.cs

Phase2Opcodes.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::Phase2Opcodes

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file Phase2Opcodes.cs

PhilipsSopho.cs File Reference

Detailed Description

Definition in file PhilipsSopho.cs

ResetSessionInfo.cs File Reference

Detailed Description

Definition in file ResetSessionInfo.cs

ROSEParseInfo.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::ROSEParseInfo

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file ROSEParseInfo.cs

RunTimeFloatLicInfo.cs File Reference

Detailed Description

Definition in file RunTimeFloatLicInfo.cs

RunTimeLicAckResp.cs File Reference

Detailed Description

Definition in file RunTimeLicAckResp.cs

RunTimeLicCheckInReq.cs File Reference

Detailed Description

Definition in file RunTimeLicCheckInReq.cs

RunTimeLicCheckOutReq.cs File Reference

Detailed Description

Definition in file RunTimeLicCheckOutReq.cs

RunTimeLicCheckOutResp.cs File Reference

Detailed Description

Definition in file RunTimeLicCheckOutResp.cs

RunTimeLicPIDUpdateReq.cs File Reference

Detailed Description

Definition in file RunTimeLicPIDUpdateReq.cs

SamsungSCM.cs File Reference

Classes

- struct Com::Objsys::Csta::Devices::SamsungSCM

Namespaces

- struct Com::Objsys::Csta::Devices

Detailed Description

Definition in file SamsungSCM.cs

SiemensCap.cs File Reference

Detailed Description

Definition in file SiemensCap.cs

SiemensHicom300.cs File Reference

Detailed Description

Definition in file SiemensHicom300.cs

SiemensHipath3000p2.cs File Reference

Classes

- struct Com::Objsys::Csta::Devices::SiemensHipath3000p2

Namespaces

- struct Com::Objsys::Csta::Devices

Detailed Description

Definition in file SiemensHipath3000p2.cs

SiemensHipath3000p3.cs File Reference

Detailed Description

Definition in file SiemensHipath3000p3.cs

SiemensHipath4000.cs File Reference

Detailed Description

Definition in file SiemensHipath4000.cs

SiemensRealitis.cs File Reference

Detailed Description

Definition in file SiemensRealitis.cs

SocketState.cs File Reference

Classes

- struct Com::Objsys::Csta::Phase2::SocketState

Namespaces

- struct Com::Objsys::Csta::Phase2

Detailed Description

Definition in file SocketState.cs

TadiranCoral.cs File Reference

Detailed Description

Definition in file TadiranCoral.cs

uaSIPInvite.cs File Reference

Detailed Description

Definition in file uaSIPInvite.cs

UnifyOpenscape4000BER.cs File Reference

Detailed Description

Definition in file UnifyOpenscape4000BER.cs

UnifyOpenscapeVoice.cs File Reference

Detailed Description

Definition in file UnifyOpenscapeVoice.cs

UnifyOpenscapeX5.cs File Reference

Detailed Description

Definition in file UnifyOpenscapeX5.cs

Version.cs File Reference

Detailed Description

Definition in file Version.cs

VodiaSNOMOne.cs File Reference

Detailed Description

Definition in file VodiaSNOMOne.cs

XMLParseInfo.cs File Reference

Detailed Description

Definition in file XMLParseInfo.cs