

Inhaltsverzeichnis

Was leisten die ISDN-Controls für Delphi

Generelles zum Thema ISDN

Das Basisobject TISDNControl

Fehlermeldungen

Service-Adresse

Was leisten die ISDN-Controls für Delphi

Die ISDN-Controls stellen über das Objekt TISDNControl alle grundlegenden Funktionen für die Kommunikation mit dem CAPI-Treiber zur Verfügung. Andere Treiber werden nicht benötigt.

Eine genaue Kenntnis der CAPI-Funktionsweise ist, soweit es Datenübertragung betrifft, nicht erforderlich. Für spezielle Anwendungen, z.B. Voice-Systeme, Fax-Empfang und -Versand sind die entsprechenden Dokumentationen 1TR6 sowie das sogenannten BlueBook der ITU Pflicht. Die Funktionen die nicht die Datenübertragung betreffen in dieser Dokumentation zu erläutern, würde nicht nur den Rahmen dieses Help-Files sondern auch Ihre Plattenkapazitäten sprengen.

Für spezielle Anwendungsgebiete planen wir für die Zukunft eigenständige Komponenten die auf TISDNControl basieren und als modulares Baukastensystem entsprechend erworben werden können. Im Rahmen dieser Komponenten werden wir auf Teilbereiche der oben genannten Dokumentationen eingehen.

Für die nächsten Versionen sind zunächst Übertragungsprotokolle wie Z-Modem, X-Modem, Y-Modem geplant. Der nächste Schritt geht dann in Richtung Fax-Empfang und Versand, gefolgt von einer Anruferbeantworterkomponente. Sind diese Komponenten fertiggestellt, werden wir versuchen für bestimmte Fachbereiche wie z.B. InterNet und FidoNet Komponenten zu entwickeln.

Sie sehen also das für die Zukunft durchaus noch eine Menge Arbeit vor uns liegt.

Da auch Programmierer nicht allwissend sind, möchten wir Sie bitten, sich mit uns in Verbindung zu setzen, insofern Sie Anregungen oder Verbesserungsvorschläge haben.

TISDNControl

[Eigenschaften](#)

[Methoden](#)

[Ereignisse](#)

Unit

[RDS_ISDN](#)

Beschreibung

Die Komponente TISDNControl stellt die Basiskomponente für alle Zugriffe auf die [CAPI.DLL](#) dar. Jegliche Ein-/Ausgabeoperationen die den ISDN-Controller betreffen, werden über diese Komponente erledigt.

Die wichtigste Eigenschaft des Objektes ist die Eigenschaft [CapiDir](#) in der der Pfad zu der CAPI.DLL festgelegt wird.

Über die Eigenschaften [MsgCount](#), [MaxB3Connection](#), [MaxB3DataBlocks](#) und [MaxB3DataLen](#) definieren Sie die grundlegenden Informationen zur Steuerung der [CAPI.DLL](#). [MaxB3DataLen](#) und [B2DataLength](#) sollten identische Werte aufweisen, da es ansonsten zu Problemen bei der Datenübertragung führen kann.

Die Protokollsicherung für die zustandekommenden Verbindungen, wird über die beiden Eigenschaften [B2Protocoll](#) und [B3Protocoll](#) festgelegt. Welche Kombinationen hier einzustellen sind, entnehmen Sie bitte den entsprechenden Dokumentationen [1TR6](#) sowie dem sogenannten [BlueBook](#) der ITU (Für Datenkommunikation belassen Sie bitte die Standardeinstellungen des Controls).

Gleiches gilt für die Eigenschaften [InService](#) und [OutService](#), über die Sie die unterstützen Dienste bzw. den gewünschten Dienst einstellen können (z.B. Datenübertragung, Fax, BTX).

Welche [EAZ](#) beim Wählen benutzt werden soll, legen Sie über [EAZOut](#) fest. Für die Eingangüberwachung können Sie in [EAZIn](#) eine Mehrfachauswahl festlegen (z.B. Überwachung mehrerer EAZ, wobei EAZ 0 eine Sonderstellung beinhaltet).

Möchten Sie bestimmte Informationsdienste nutzen (z.B. bisher angefallene Gebühren), so stellen Sie diese in den beiden Eigenschaften [InInfoMask](#) und [OutInfoMask](#) ein. Kombinationen sind möglich. Je nach eingestellten Informationsdiensten werden beim Eintreffen der Informationsmitteilung die Ereignisse [OnInfCause](#), [OnInfDatum](#), [OnInfDisplay](#), [OnInfGebuehren](#), [OnInfStatus](#) oder [OnInfUserUser](#) ausgelöst.

Zugriff auf die verschiedenen Verbindungsstufen bieten die Ereignisse [OnConnection](#), [OnDisConnection](#), [OnSchicht1Connect](#), [OnSchicht2Connect](#), [OnSchicht3Connect](#) und [OnB3DisConnection](#).

Zur Laufzeit des Programmes müssen Sie zunächst die Eigenschaft [Active](#) setzen, damit TISDNControl das CAPI lädt. War die Funktion erfolgreich beinhaltet die Eigenschaft [AppID](#) eine eindeutige Applikationsnummer, die vom CAPI vergeben wurde und für weitere Korrespondenz benötigt wird, ebenso wird die Eigenschaft [Installed](#) auf True gesetzt. Als nächstes müssen Sie nun über die Eigenschaft [Signal](#) die Nachrichtenbearbeitungsroutine von TISDNControl aktivieren, indem Sie den Wert der Eigenschaft auf TRUE setzen. Nun Können Sie über die Eigenschaften [ManuFacterer](#) den Hersteller, über [SerialNumber](#) die Seriennummer der installierten CAPI.DLL, sowie über die Eigenschaft [Version](#) die Versionsnummer des geladenen CAPI abfragen. Diese Funktionen stehen ohne weitere Funktionsaufrufe zur Verfügung.

Bevor Sie den Aufbau einer Verbindung über [Connect_Request](#) starten, legen Sie in der Eigenschaft [DestAddress](#) die Zielrufnummer fest. Den eigentlichen Verbindungsaufbau übernimmt das ISDNControl, benachrichtigt Sie jedoch wie zuvor beschrieben, über die entsprechend eingestellten Methoden über den aktuellen Status.

Sollten Daten empfangen worden sein, wird die unter OnDataReceived angegebene Methode aufgerufen. Die im Puffer des CAPI stehenden Daten können Sie über GetReceivedData abrufen. Anschließend benachrichtigen Sie das CAPI über Data_B3_Resp, das der Empfangspuffer wieder geleert werden kann. **Bevor Sie Data_B3_Resp aufrufen, müssen Sie die empfangenen Daten abgearbeitet haben, da nach diesem Aufruf alle Daten wieder gelöscht werden.** Nach dem anschließend das Ereignis OnDataSent eingetreten ist, können Sie wieder Daten senden.

Eigenschaften

▶ Nur zur Laufzeit

🔑 Schlüsseleigenschaften

- 🔑 Active
- 🔑 MsgCount
- ▶ Appld
- ▶ ManuFacterer
- 🔑 B2Protocoll
- 🔑 MaxB3Connection
- 🔑 B2DataLength
- 🔑 MaxB3DataBlocks
- 🔑 B3Protocoll
- 🔑 MaxB3DataLen
- BChannel 🔑 OutInfoMask
- 🔑 CapiDir
- 🔑 OutService
- ▶ Connected ▶ OutServiceAdd
- ControllerOut ▶ ReceivedBlock
- ControllerIn ▶ RequestedService
- 🔑 DestAddress
- ▶ RequestedEAZ
- 🔑 EAZIn
- ▶ Schicht1Connected
- 🔑 EAZOut
- ▶ Schicht2Connected
- 🔑 InInfoMask
- ▶ Schicht3Connected
- 🔑 InService
- ▶ SerialNumber
- ▶ Installed
- 🔑 Signal
- 🔑 LastError
- ▶ Version

Die CAPI.DLL wird von dem Hersteller Ihrer ISDN-Karte geliefert und dient als Schnittstellenprogramm zwischen Applikationen und der Hardware. Je nach Hersteller können weitere Funktionen zur Verfügung stehen, die nicht von TISDN-Control unterstützt werden.

OnSchicht2Connect, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration


property OnSchicht2Connect:TNotifyEvent;


Beschreibung

Das Ereignis OnSchicht2Connect tritt ein, wenn eine B2-Verbindung (Ebene 2) erfolgreich aufgebaut wurde.

Methoden

Schlüsselmethoden


 Clear_Listen_Request


 Connect_Request

 Connect_Response

Data_B3_Req

Data_B3_Resp

 Disconnect_B3_Request

 Listen_Request

DTMF_Req

GetReceivedData










Get_Params_Request

Get_B3_Params_Request

Info_Request

Ereignisse

Schlüsselereignisse

	<u>OnConnection</u>		<u>OnInfDatum</u>
	<u>OnB3DisConnection</u>		<u>OnInfDisplay</u>
	<u>OnDisConnection</u>		<u>OnInfUserUser</u>
	<u>OnDataSent</u>		<u>OnInfCause</u>
	<u>OnDataReceived</u>		<u>OnInfStatus</u>
	<u>OnDTMFIn</u>		<u>OnSchicht1Connect</u>
	<u>OnHandSet</u>		<u>OnSchicht2Connect</u>
	<u>OnInComingCall</u>		
	<u>OnSchicht3Connect</u>		
	<u>OnInfGebuehren</u>		

14bändiges Werk, in dem die ISDN-Standardbeschreibungen definiert sind (D-Kanal-Beschreibungen finden Sie in 1 TR 6)

Bezugsquelle: Telekom Eurolab Roland
Dokumentenservice
Postfach 9100
55541 Bad Kreuznach
Tel.: 06 71 / 83 33 - 0
Fax.: 06 71 / 83 33 - 83

Preis ca. 1000,- DM für alle Empfehlungen zusammen, die jedoch auch einzeln zu beziehen sind.

Beschreibung des nationalen D-Kanal-Protokolls (Referenzierung durch CAPI-Dokumentation).

Bezugsquelle: FTZ Darmstadt
Z 45-DrV-11
Postfach 100 003
64276 Darmstadt
Tel.: 0 61 51 / 83 83 62
Fax.: 0 61 51 / 83 14 27

Preis ca. 250,- DM

MsgCount, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property MsgCount:Word;

Beschreibung

Die Eigenschaft MsgCount legt die Anzahl der CAPI-Nachrichten fest, die gespeichert werden können, bevor ein Datenverlust eintritt. Je höher MsgCount gesetzt wird, desto unwarscheinlicher ist ein Datenverlust. [Der Vorgabewert beträgt: 10.](#)

Beachten Sie bitte, das anhand der Formel: $(180 * \text{MsgCount}) + (\text{MaxB3Connection} * \text{MaxB3DataBlocks} * \text{MaxB3DataLen})$ dynamischer Speicher zur Laufzeit reserviert wird. Sind die entsprechende Werte zu hoch eingestellt, kann das Programm evtl. nicht starten und bricht mit einer Fehlermeldung ab.

MaxB3Connection, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property MaxB3Connection:Word;

Beschreibung

Die Eigenschaft MaxB3Connection legt die Anzahl der B3-Verbindungen fest, die Ihre Anwendung über alle geöffneten B1-Verbindungen herstellen kann. [Der Vorgabewert beträgt: 1.](#)

Beachten Sie bitte, dass anhand der Formel: $(180 * \text{MsgCount}) + (\text{MaxB3Connection} * \text{MaxB3DataBlocks} * \text{MaxB3DataLen})$ dynamischer Speicher zur Laufzeit reserviert wird. Sind die entsprechenden Werte zu hoch eingestellt, kann das Programm evtl. nicht starten und bricht mit einer Fehlermeldung ab.

MaxB3DataBlocks, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property MaxB3DataBlocks:Word;

Beschreibung

Die Eigenschaft MaxB3DataBlocks legt die Anzahl von B3-Datenblöcken fest, die das CAPI speichern kann, bevor ein datenverlust eintritt. [Der Vorgabewert beträgt: 7.](#)

Beachten Sie bitte, das anhand der Formel: $(180 * \text{MsgCount}) + (\text{MaxB3Connection} * \text{MaxB3DataBlocks} * \text{MaxB3DataLen})$ dynamischer Speicher zur Laufzeit reserviert wird. Sind die entsprechende Werte zu hoch eingestellt, kann das Programm evtl. nicht starten und bricht mit einer Fehlermeldung ab.

MaxB3DataLen, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property MaxB3DataLen:Word;

Beschreibung

Die Eigenschaft MaxB3DataLen legt die maximale Größe eines B3-Datenblocks fest. [Der Vorgabewert beträgt: 2048.](#)

Dieser Wert wird automatisch auf den in der Eigenschaft B2DataLength festgelegten Wert eingestellt.

Beachten Sie bitte, dass anhand der Formel: $(180 * \text{MsgCount}) + (\text{MaxB3Connection} * \text{MaxB3DataBlocks} * \text{MaxB3DataLen})$ dynamischer Speicher zur Laufzeit reserviert wird. Sind die entsprechenden Werte zu hoch eingestellt, kann das Programm evtl. nicht starten und bricht mit einer Fehlermeldung ab.

2048 ist auch gleichzeitig der größtmögliche Wert der derzeit vom CAPI unterstützt wird. Geben Sie einen größeren Wert ein, gibt es zur Laufzeit eine Fehlermeldung, insofern das installierte CAPI in der Version bis 2.97 vorliegt.

B2DataLength, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property B2DataLength:Word;

Beschreibung

Die Eigenschaft B2DataLength legt die Größe eines B2-Datenblocks fest. [Der Vorgabewert beträgt: 2048.](#) Dieser Wert wird automatisch auf den in der Eigenschaft MaxB3DataLen festgelegten Wert eingestellt.

2048 ist auch gleichzeitig der größtmögliche Wert der derzeit vom CAPI unterstützt wird. Geben Sie einen größeren Wert ein, gibt es zur Laufzeit eine Fehlermeldung, insofern das installierte CAPI in der Version bis 2.97 vorliegt.

Generelles zum Thema ISDN

Wie schon im Kapitel Was leisten die ISDN-Controls angesprochen, würde eine Erläuterung der genauen Funktionsweise des ISDN und der Programmierung des CAPI den Rahmen des Möglichen sprengen. Dieses Kapitel soll Ihnen den generellen Ablauf der CAPI-Programmierung mit TISDNControl näherbringen.

Die wichtigste Eigenschaft der Komponente

Die wichtigste Eigenschaft der Komponente TISDNControl ist die Eigenschaft CapiDir. In Ihr legen Sie den Pfad zu der CAPI.DLL fest. Kann die DLL nicht gefunden werden, funktioniert das gesamte Programm nicht. Sollten Sie ein Programm mit TISDNControl entwickeln und dieses ausliefern, empfiehlt sich immer die Benutzung einer INI-Datei, in die der Anwender den entsprechenden Pfad eintragen kann.

Eine aktive Verbindung aufbauen

Um eine aktive Verbindung herzustellen, müssen Sie zunächst die entsprechenden Eigenschaften der Komponente TISDNControl einstellen. Für Datenübertragungszwecke können Sie die für Eigenschaften B2DataLength, B2Protocoll, B3Protocoll, MaxB3Connection, MaxB3DataBlocks, MaxB3DataLen, MsgCount und OutService die voreingestellten Werte belassen.

Zur Laufzeit des Programmes müssen Sie zunächst die Eigenschaft Active setzen, damit TISDNControl das CAPI lädt. War die Funktion erfolgreich beinhaltet die Eigenschaft AppID eine eindeutige Applikationsnummer, die vom CAPI vergeben wurde und weitere Korrespondenz benötigt wird, ebenso wird die Eigenschaft Installed auf True gesetzt.

Bevor Sie nun auf die CAPI-Nachrichtenfunktionen zugreifen können, muß der Wert der Eigenschaft Signal auf TRUE gesetzt werden. Dies aktiviert die Nachrichtenbearbeitungsroutine von TISDNControl. **Ist diese Eigenschaft nicht aktiviert ist ein weiteres Arbeiten mit dem CAPI nicht möglich.**

Nun können Sie über die Eigenschaften ManuFacterer den Hersteller, über SerialNumber die Seriennummer der installierten CAPI.DLL (sofern vom Hersteller vorgesehen) sowie über die Eigenschaft Version die Versionsnummer des geladenen CAPI abfragen. Diese Funktionen stehen ohne weitere Funktionsaufrufe zur Verfügung.

Der letzte Schritt vor dem Aufbau einer aktiven Verbindung besteht nun darin, über die Eigenschaft DestAddress die gewünschte Zielrufnummer festzulegen. Ist dies erfolgt, rufen Sie lediglich Connect_Request auf. Den eigentlichen Verbindungsaufbau übernimmt das ISDNControl für Sie. Haben Sie die verschiedenen Ereignisbehandlungsroutinen aktiviert, benachrichtigt Sie TISDNControl nun nach jedem erfolgreichen Sicherungsschichtaufbau. über die entsprechend eingestellten Methoden.

Über die verschiedenen Stufen des Verbindungsaufbaus informieren Sie die Ereignisse OnSchicht1Connect, OnSchicht2Connect, OnSchicht3Connect. Erst nach dem Eintreten des Ereignisses OnConnection wurde die Verbindung erfolgreich aufgebaut.

Sie können nun über die Methode Data_B3_Reg versenden. Über das Ereignis OnDataSent erfahren Sie, ob der zu übertragende Datenblock von der Gegenseite empfangen wurde. OnDataReceived teilt Ihnen mit, das Daten von der Gegenstelle gesendet wurden. Rufen Sie die Methode GetReceivedData auf um einen Zeiger auf den empfangenen Datenblock zu erhalten. Haben Sie den empfangenen Datenlock abgearbeitet teilen Sie dies dem CAPI über Data_B3_Resp mit.

Eine Verbindung abbauen

Um eine Verbindung zu beenden, rufen Sie die Methode Disconnect_B3_Request auf. Alle weiteren Schritte werden von TISDNControl erledigt.

Sollte die Verbindung von der Gegenstelle abgebaut werden werden, erhalten Sie hierüber die

Information über die Eigenschaft OnB3DisConnection. Dieses Ereignis wird immer dann ausgelöst, wenn eine bestehende Ebene 3 Verbindung beendet wird. Erst wenn das Ereignis OnDisConnection eingetreten ist, wurde die gesamte Verbindung (Schicht 3, Schicht 2 und Schicht 1) abgebaut.

Service-Adresse

Sollten Sie Fragen oder Anregungen haben, erreichen Sie uns unter:

RDS Rother Daten Service
Pellwormstr. 13
45665 Recklinghausen

Tel.: 0 23 61 / 4 31 34

Fax.: 0 23 61 / 9 40 14

BBS: 0 23 61 / 9 40 21 (ISDN und Analog 2.400bps bis 28.800bps)

Fido: 2:2444/8022

TGebuehrEvent = **Procedure**(Sender:TObject;Gebuehr:Longint) **of object**;

CapiDir, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property CapiDir:String;

Beschreibung

Die Eigenschaft CapiDir legt den Pfad für die CAPI.DLL fest. [Der Vorgabewert ist: C:\TELESCOM\](#)

Da die CAPI.DLL vom Objekt TISDNControl zur Laufzeit dynamisch geladen wird, muß die DLL in diesem Verzeichnis vorhanden sein, da ansonsten das Objekt nicht funktioniert.

B2Protocoll, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property B2Protocoll:TB2Protocol;

Beschreibung

Die Eigenschaft B2Protocoll legt das Protokoll für die zweite Ebene (Sicherheitsschicht) fest. [Der Vorgabewert ist: X75_SLP](#)

B2Protocoll ist eine Eigenschaft vom Typ TB2Protocoll mit vorgegebenen Sicherungsprotokollen. Zur Auswahl stehen hier:

X75_SLP
SNA_SDLC

Transparent_HDLC
X75_BTX

Bit_Transparent
Rec_BitTransparent

Für Datenübertragung ist die Voreinstellung *X75_SLP* zu wählen.

OutService, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property OutService:TService;

Beschreibung

Die Eigenschaft OutService legt den gewünschten Dienst für einen ausgehenden Anruf fest. [Der Vorgabewert ist: Dateneübertragung](#)

OutService ist eine Eigenschaft vom Typ TService mit vorgegebenen Dienstkennungen. Zur Auswahl stehen hier:

BildTelefon
TeleFax_Grp4
Teletex_64
BTX_NeuerStandard

Telefon
Bildschirmtext
Mixed_Mode

AB_Dienste
Dateneübertragung
Fernwirken

X21_Dienste
X25_Dienste
Grafiktelefon

B3Protocoll, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property B3Protocoll:TB3Protocol;

Beschreibung

Die Eigenschaft B3Protocoll legt das Protokoll für die dritte Ebene (Netzwerkschicht) fest. [Der Vorgabewert ist: Transparent](#)

B3Protocoll ist eine Eigenschaft vom Typ TB3Protocol mit vorgegebenen Sicherungsprotokollen. Zur Auswahl stehen hier:

T70_NL

ISO_8208

T_90

Transparent

Für Datenübertragung ist die Voreinstellung *Transparent* zu wählen, da die Sicherung bereits über das Protokoll auf Ebene 2 erfolgt (B2Protocoll).

InService, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property InService:TInService;

Beschreibung

Die Eigenschaft InService legt die gewünschten Dienste für eingehende Anrufe fest. [Der Vorgabewert ist: Dateneübertragung](#)

InService ist eine Eigenschaft vom Typ TInService mit vorgegebenen Dienstekennungen. Zur Auswahl stehen hier:

<i>BildTelefon</i>	<i>Telefon</i>	<i>AB_Dienste</i>	<i>X21_Dienste</i>
<i>TeleFax_Grp4</i>	<i>Bildschirmtext</i>	<i>Dateneübertragung</i>	<i>X25_Dienste</i>
<i>Teletex_64</i>	<i>Mixed_Mode</i>	<i>Fernwirken</i>	<i>Grafiktelefon</i>
<i>BTX_NeuerStandard</i>			

Im Gegensatz zu OutService können Sie bei InService verschiedene Dienstekennungen anwählen. Erfolgt ein eingehender Anruf, wird geprüft ob der entsprechende Dienst zur Annahme ausgewählt wurde.

Endgeräte Auswahl Ziffer

EAZOut, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property EAZOut:TEAZOut;

Beschreibung

Die Eigenschaft EAZOut legt die EAZ die bei ausgehenden Anrufen benutzt werden soll. [Der Vorgabewert ist: EAZ0](#)

EAZOut ist eine Eigenschaft vom Typ TEAZOut mit vorgegebenen Endgeräteauswahlziffern von EAZ0 bis EAZ9.

Anhand der EAZ erkennt das CAPI die zu übermittelnde Rufnummer.

EAZIn, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property EAZIn:TEAZInSet;

Beschreibung

Die Eigenschaft EAZIn legt die zu überwachenden EAZ fest. [Der Vorgabewert ist: \[EAZ0\]](#)

EAZIn ist eine Eigenschaft vom Typ TEAZInSet mit vorgegebenen Endgeräteauswahlziffern von EAZ0 bis EAZ9.

Geht ein Anruf auf einer der selektierten EAZ ein, löst dies ein OnIncomingCall-Ereignis aus.

InInfoMask, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property InInfoMask:TInfoMsgSet;

Beschreibung

Die Eigenschaft InInfoMask legt die zu übermittelnden Informationen bei eingehenden Anrufen fest.

Der Vorgabewert ist: [iGebuehr, iDatum, iDisplay, iUser, iCause, iStatus]

InInfoMask ist eine Eigenschaft vom Typ TInfoMsgSet mit vorgegebenen Informationskennungen.

Trifft eine der selektierten Informationsnachrichten vom CAPI ein, werden die entsprechend die Ereignisse OnInfGebuehren, OnInfDatum, OnInfDisplay, OnInfUserUser, OnInfCause, OnInfStatus ausgelöst.

OutInfoMask, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property OutInfoMask:TInfoMsgSet;

Beschreibung

Die Eigenschaft OutInfoMask legt die zu übermittelnden Informationen bei ausgehenden Anrufen fest.

Der Vorgabewert ist: [iGebuehr, iDatum, iDisplay, iUser, iCause, iStatus]

OutInfoMask ist eine Eigenschaft vom Typ TInfoMsgSet mit vorgegebenen Informationskennungen.

Trifft eine der selektierten Informationsnachrichten vom CAPI ein, werden die entsprechend die Ereignisse OnInfGebuehren, OnInfDatum, OnInfDisplay, OnInfUserUser, OnInfCause, OnInfStatus ausgelöst.

Active, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Active:Boolean;

Beschreibung

Weisen Sie der Eigenschaft Active zur Laufzeit des Programmes den Wert TRUE zu, damit TISDNControl die CAPI.DLL in den Speicher lädt, sofern diese noch nicht von einer anderen Anwendung geladen wurde.

Nach erfolgreichem Laden der DLL beinhaltet die Eigenschaft AppID die vom CAPI vergebene Applikationsnummer und der Wert der Eigenschaft Installed ist TRUE.

AppID, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property AppID:Word;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft AppID enthält die vom CAPI vergebene Applikationsnummer.

Installed, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Installed:Boolean;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft Installed enthält TRUE wenn das CAPI erfolgreich geladen wurde.

Schicht1Connected, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Schicht1Connected:Boolean;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft Schicht1Connected enthält TRUE wenn eine D-Kanal-Verbindung erfolgreich hergestellt werden konnte. Eine Benachrichtigung über den erfolgreichen Aufbau einer D-Kanal-Verbindung erhalten Sie über das Ereignis OnSchicht1Connect.

Schicht2Connected, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Schicht2Connected:Boolean;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft Schicht2Connected enthält TRUE wenn eine B2-Verbindung erfolgreich hergestellt werden konnte. Eine Benachrichtigung über den erfolgreichen Aufbau einer B2-Verbindung erhalten Sie über das Ereignis OnSchicht2Connect.

Schicht3Connected, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Schicht3Connected:Boolean;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft Schicht3Connected enthält TRUE wenn eine B3-Verbindung erfolgreich hergestellt werden konnte. Eine Benachrichtigung über den erfolgreichen Aufbau einer B3-Verbindung erhalten Sie über das Ereignis OnSchicht3Connect.

ManuFacterer, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property ManuFacterer:String;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft ManuFacterer enthält den Hersteller der CAPI.DLL.

Version, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Version:String;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft Version enthält die Versionsnummer der CAPI.DLL.

SerialNumber, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property SerialNumber:String;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft SerialNumber enthält die Seriennummer der CAPI.DLL. Diese Funktion wird nicht von jedem Hersteller unterstützt und kann gegebenenfalls eine leere Zeichenkette zurückliefern.

DestAddress, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property DestAddress:String;

Beschreibung

Weisen Sie der Eigenschaft DestAddress zur Laufzeit des Programmes die gewünschte Zielrufnummer zu.

Fehlermeldungen

Laufzeit-Fehlermeldungen können in der Eigenschaft LastError der Komponente TISDNControl abgefragt werden.

Fehlermeldungen die beim Setzen der Eigenschaft Active auftreten können (Versuch die CAPI.DLL zu laden)

Fehlernr.	Beschreibung
0	Zu wenig Speicher, ausführbare Datei defekt oder ungültige Relokationen
2	Datei nicht gefunden
3	Pfad nicht gefunden
5	Versuch, eine Task dynamisch einzubinden, oder ein Share- oder Netzwerkfehler
6	Bibliothek benötigt getrennte Datensegmente für jede Task
8	Zu wenig Speicher um die Anwendung zu starten
10	Falsche Windows-Version

Fehlermeldungen die beim Zugriff auf Warteschlangenfunktionen auftreten können (interne Fehler der TISDNControl)

Fehlernr.	Beschreibung
11	nicht genügend Arbeitsspeicher vorhanden (Laden von CAPI.DLL)
12	Nachrichtenwarteschlange momentan in Benutzung
13	Funktion nicht möglich da keine Verbindung aktiv.

Fehlermeldungen beim internen Zugriff auf das CAPI (CAPI-Fehler)

Fehlernr.	Beschreibung
\$1002	falsche Applikationsnummer
\$1003	Nachricht zu klein oder Nachrichtennummer falsch codiert
\$1004	falsches Command oder Sub-Command in der Nachricht
\$1005	Nachrichtenwarteschlange ist voll
\$1007	mindestens eine Nachricht ist durch den Überlauf der Warteschlange verloren'
\$2001	Falscher Controller
\$2002	Falschen PLCI angegeben
\$2003	unbekannten NCCI angegeben
\$3101	B-Kanal falsch codiert
\$3102	Info-Mask falsch codiert
\$3103	Serviced-EAZ-Mask falsch codiert
\$3104	Serviced-SI-Mask falsch codiert
\$3105	Falsches B2-Protokoll gewählt
\$3106	DLPD falsch
\$3107	Falsches B3-Protokoll gewählt
\$3108	NCPD falsch
\$3109	NCPI falsch codiert
\$310A	Flags falsch codiert

\$3202	Konflikt zwischen Anwendugen
\$3204	PLCI nicht aktiv
\$3205	NCCI nicht aktiv
\$3206	B2-Protokoll wird nicht unterstützt
\$3207	Umschalten des B2-Protokolls in diesem Zustand nicht möglich
\$3208	B3-Protokoll wird nicht unterstützt
\$3209	Umschalten des B3-Protokolls in diesem Zustand nicht möglich
\$320D	Nicht unterstützte Datenlänge
\$3301	Fehler beim Aufbau D-Kanal Ebene 1
\$3302	Fehler beim Aufbau D-Kanal Ebene 2
\$3305	Abbruch D-Kanal Ebene 1 (von Gegenstelle)
\$3306	Abbruch D-Kanal Ebene 2 (von Gegenstelle)
\$3435	Kein Anschluß unter dieser Nummer
\$3438	Rufnummer der Gegenstelle hat sich geändert
\$343A	Gegenstelle nimmt nicht ab
\$343B	Gegenstelle besetzt
\$343E	Anruf von Gegenstelle zurückgewiesen

RequestedService, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property RequestedService:TService;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft RequestedService enthält bei einem eingehenden Anruf auf einer der selektierten EAZ, den vom Anrufer gewünschten Dienst.

RequestedService ist eine Eigenschaft vom Typ TService mit vorgegebenen Dienstkennungen. Möglichkeiten sind hier:

BildTelefon
TeleFax_Grp4
Teletex_64
BTX_NeuerStandard

Telefon
Bildschirmtext
Mixed_Mode

AB_Dienste
Datenuebertragung
Fernwirken

X21_Dienste
X25_Dienste
Grafiktelefon

Signal, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Signal:Boolean;

Beschreibung

Weisen Sie der Eigenschaft Signal zur Laufzeit des Programmes den Wert TRUE zu, damit TISDNControl die Bearbeitung der CAPI-Nachrichten beginnt.

Ist Signal nicht gesetzt worden, erfolgt keine Bearbeitung der CAPI-Nachrichten. Zugriffe auf das CAPI sind somit nicht möglich.

Connected, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property Connected:Boolean;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft Connected enthält TRUE, wenn eine aktive Verbindung besteht.

ReceivedBlock, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property ReceivedBlock:Byte;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft ReceivedBlock enthält die Nummer des zuletzt vom CAPI empfangenen B3-Datenblocks.

LastError, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property LastError:Word;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft LastError enthält die zuletzt aufgetretene Fehlermeldung.

RequestedEAZ, Eigenschaft

Verwendung

TISDNControl, Komponente

Deklaration

property RequestedEAZ:Byte;

Beschreibung

Nur zur Laufzeit und nur Lesezugriff. Die Eigenschaft RequestedEAZ enthält bei einem eingehenden Anruf auf einer der selektierten EAZ, die vom Anrufer gewählte EAZ.

TInComEvent = **Procedure**(Sender:TObject;Service,EAZ:Byte;Anrufer,ServiceStr:String) **of object**;

OnSchicht1Connect, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnSchicht1Connect:TNotifyEvent;

Beschreibung

Das Ereignis OnSchicht1Connect tritt ein, wenn nach einem Connect_Request eine D-Kanal-Verbindung (Ebene 1) erfolgreich aufgebaut wurde.

OnSchicht3Connect, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnSchicht3Connect:TNotifyEvent;

Beschreibung

Das Ereignis OnSchicht3Connect tritt ein, wenn eine B3-Verbindung (Ebene 3) erfolgreich aufgebaut wurde.

OnConnection, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnConnection:TNotifyEvent;

Beschreibung

Das Ereignis OnConnection tritt ein, nachdem die Verbindungen auf Ebene 1 bis Ebene 3 erfolgreich hergestellt werden konnten und nun eine gesicherte Datenübertragung möglich ist. Erst wenn dieses Ereignis eingetreten ist, können Sie davon ausgehen, dass die Verbindung erfolgreich aufgebaut wurde.

OnB3DisConnection, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnB3DisConnection:TNotifyEvent;

Beschreibung

Das Ereignis OnB3DisConnection tritt ein, wenn eine bestehende Ebene-3 Verbindung abgebrochen wird. Dieses Ereignis tritt z.B. dann ein, wenn die Gegenstelle den Verbindungsabbau einleitet.

OnDisConnection, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnDisConnection:TNotifyEvent;

Beschreibung

Das Ereignis OnDisConnection tritt ein, wenn alle bestehenden Verbindungen auf der Ebene-1 bis Ebene-3 abgebaut wurden. Erst nach diesem Ereignis ist ein erneuter Verbindungsaufbau möglich.

OnDataSent, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnDataSent:TNotifyEvent;

Beschreibung

Das Ereignis OnDataSent tritt ein, nachdem ein zuvor mit Data_B3_Req gesendeter Datenblock vom CAPI erfolgreich verschickt wurde. Sie sollten neue Daten erst nach dem Eintreten dieses Ereignisses versenden.

OnDataReceived, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnDataReceived:TNotifyEvent;

Beschreibung

Das Ereignis OnDataReceived tritt ein, wenn vom CAPI ein Datenblock empfangen wurde.

Den empfangenen Block können Sie über GetReceivedData referenzieren und danach abarbeiten. Erst nach dem Erhalt einer Data_B3_Resp-Nachricht kann das CAPI weitere Datenblöcke empfangen. Sobald Data_B3_Resp aufgerufen wird, wird der empfangene Datenblock aus dem Speicher gelöscht, Sie sollten also zuvor sicherstellen, dass empfangene Daten abgearbeitet wurden.

OnInComingCall, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnInComingCall:TInComEvent;

Beschreibung

Das Ereignis OnInComingCall tritt ein, wenn auf einer der selektierten EAZ ein Anruf eingeht.

Im Parameter Service steht die Byte-Kennung für den eingehenden Service. Der Parameter EAZ enthält die vom Anrufenden gewählte EAZ (Telefonnummer). In Anrufer wird, sofern übermittelt, die Nummer des Anrufers übergeben. Der Parameter ServiceStr enthält letztlich den eingehenden Service als Zeichenkette.

OnInfGebuehren, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnInfGebuehren:TGebuehrEvent;

Beschreibung

Das Ereignis OnInfGebuehren tritt ein, wenn eine Gebühreninformation vom CAPI eintrifft. Im Parameter Gebuehr werden die, bis zum Zeitpunkt des Ereignisses, aufgetretenen Gebühren in der Einheit Pfennig übergeben. Eine angefallene von Gebühr von 1,34 DM würde demzufolge 134 im Parameter Gebuehr beinhalten.

OnInfDatum, Ereignis

Verwendung

TISDNControl, Komponente

Deklaration

property OnInfDatum:TStringEvent;

Beschreibung

Das Ereignis OnInfDatum tritt ein, wenn eine Information vom Typ Datum/Uhrzeit vom CAPI eintrifft. Im Parameter Puffer wird die entsprechende Information übergeben.

OnInfStatus

Verwendung

TISDNControl, Komponente

Deklaration

property OnInfStatus:TStringEvent;

Beschreibung

Das Ereignis OnInfStatus tritt ein, wenn eine Information vom Typ Status vom CAPI eintrifft. Im Parameter Puffer wird die Rufnummer der Gegenstelle übergeben.

Connect_Request, Methode

Verwendung

TISDNControl, Komponente

Deklaration

procedure Connect_Request;

Beschreibung

Rufen Sie die Methode Connect_Request auf um eine aktive Verbindung mit dem in der Eigenschaft DestAddress angegebenen Teilnehmer aufzubauen.

Connect_Response, Methode

Verwendung

TISDNControl, Komponente

Deklaration

procedure Connect_Response(Accept:Boolean);

Beschreibung

Rufen Sie die Methode Connect_Response auf, wenn Sie einen eingehenden Anruf entgegennehmen oder ablehnen möchten. Übergeben Sie im Parameter *Accept* TRUE so wird der Anruf entgegengenommen, bei FALSE hingegen abgewiesen.

Üblicherweise deklarieren Sie zunächst eine Ereignisbehandlungsroutine für das Ereignis OnIncomingCall und reagieren innerhalb der deklarierten Methode mit Connect_Response..

Clear_Listen_Request, Methode

Verwendung

TISDNControl, Komponente

Deklaration

procedure Clear_Listen_Request;

Beschreibung

Rufen Sie die Methode Clear_Listen_Request auf, um eine über Listen_Request eingeleitete EAZ-Überwachung zu beenden.

Sollten Sie eine eingeleitete Überwachung nicht über Clear_Listen_Request beenden, so kann es bei einem erneuten Zugriff auf das CAPI zu einem Systemhalt kommen.

TStringEvent = **Procedure**(Sender:TObject;Puffer:String) **of object**;

Listen_Request

Verwendung

TISDNControl, Komponente

Deklaration

procedure Listen_Request;

Beschreibung

Rufen Sie die Methode Listen_Request auf, um eine EAZ-Überwachung zu beginnen.

Beendet wird die Überwachung über die Methode Clear_Listen_Request. Sollten Sie eine eingeleitete Überwachung nicht über Clear_Listen_Request beenden, so kann es bei einem erneuten Zugriff auf das CAPI zu einem systemhalt kommen.

Disconnect_B3_Request, Methode

Verwendung

TISDNControl, Komponente

Deklaration

procedure Disconnect_B3_Request;

Beschreibung

Rufen Sie die Methode Disconnect_B3_Request auf um eine B3-Verbindung zu beenden.

Bei der Verbindung kann es sich sowohl um eine aktive (Aufbau erfolgte über Connect_Request) als auch um eine passive Verbindung (Aufbau erfolgte über Connect_Response) handeln.

Data_B3_Req, Methode

Verwendung

TISDNControl, Komponente

Deklaration

procedure Data_B3_Req(DataPtr:Pointer;BlockNr:Byte;Len:Word);

Beschreibung

Über die Methode Data_B3_Req können Sie Daten über das CAPI versenden.

Der Parameter *DataPtr* beinhaltet einen Zeiger auf den zu versendenden Datenblock.

Als *BlockNr* übergeben Sie eine eindeutige Blocknummer (dies ist dann relevant sobald sie mehrerer Datenblöcke senden möchten, ohne auf ein das Ereignis OnDataSent zu warten). Arbeiten Sie nur mit einem Block und wenn anschließend auf das OnDataSent Ereignis können sie einfach 1 als *BlockNr* übergeben.

Im *Len* geben Sie die Länge des zu versendenden Datenblocks an.

Data_B3_Resp, Methode

Verwendung

TISDNControl, Komponente

Deklaration

```
procedure Data_B3_Resp(BlockNr:Byte);
```

Beschreibung

Über die Methode `Data_B3_Resp` teilen Sie dem CAPI mit, das der Empfangspuffer geleert werden kann.

Über den Parameter *BlockNr* übergeben Sie die Blocknummer des zuvor empfangenen Datenblocks. Die Blocknummer steht in der Eigenschaft ReceivedBlock. Dieser Parameter ist nicht zwingend. Übergeben Sie einfach 0, wenn Sie Block-Funktion nicht nutzen wollen.

Nach Aufruf dieser Methode wird der im Speicher befindliche Datenblock gelöscht.

Die empfangenen Daten werden über die Methode `GetReceivedData` geholt.

GetReceivedData, Methode

Verwendung

TISDNControl, Komponente

Deklaration

procedure GetReceivedData(DataPtr:Pointer;Var Len:Word);

Beschreibung

Über die Methode GetReceivedData erhalten Sie Zugriff auf den Empfangspuffer des CAPI.

In DataPtr übergeben Sie einen Zeiger auf einen in Ihrer Anwendung zur Verfügung stehenden Puffer. Der Parameter Len enthält nach Aufruf der Methode die Länge des Datenpuffers.

