

TAPIMaster® Programmierhandbuch

© 2020 Tino Kasubke

Inhaltsverzeichnis

Inhalt	1
1 Allgemeines	2
2 Client-Schnittstelle	3
Befehlsreferenz	4
.NET	5
CtiConvert	6
ConvertToCanonical	7
ExpandInternNumber	8
GetDateTime	9
GetFormatDateTime	10
CtiLogin	12
AssistedTelephony	14
AutoAnswer	15
AutoConnect	16
AutoLogin	17
BeforeClose	18
CheckConnection	19
ClientActive	20
ConfigFileName	21
HotKey	22
Login	23
Logoff	24
NetworkConfigDialog	25
OnClientClose	26
OnConnectionState	27
OnLicenseInfo	28
OnLocaleInfo	29
OnLogin	30
OnMenuCommand	31
OnRaiseClient	32
OnShowPhoneList	33
OwnName	34
OwnNumber	35
Phonelcon	36
Port	37
PrepareRestart	38
QuickMouseDial	39
SaveSettings	40
ServerName	41
SetConnectionIconOptions	42
SetMainWindow	43
ShowIcon	44
ShowInfoDialog	45
ShowMenu	46
StartClient	47
Messages	48
ChatEnd	49

ChatMessage	50
ChatStart	51
OnChatMessage.....	52
OnChatStart	53
OnChatStop	54
OnDatabaseToUser.....	55
OnServerMessage.....	56
OnUserToUser	57
SendUserToUser.....	58
UserSendToDataBase.....	59
SQLTool	60
Begin	60
End	62
getDbBook	63
Login	64
Next	65
OnDbBookRecord.....	66
OnSQLData	67
OnSQLEnd	68
OnSQLLogin	69
Telephony	70
AddCallToConference.....	72
AddNumberToConference.....	73
Alternate	74
Answer	75
DialKey	76
DialSpecialKey	77
DropKey	78
DropSpecialKey.....	79
ForwardingDest.....	80
ForwardingState.....	81
Hangup	82
Hold	83
LastDialedNumber.....	84
LineReset	85
MakeCall	86
OnAcdCancelClient.....	87
OnCallNumber	88
OnCallReminder	89
OnCallState	90
OnConfNotify	91
OnForwardingState.....	92
OnGroupMember.....	93
OnIdentity	94
OnObjectOfRedirect.....	95
OnProtocolData	96
OnSendAddress.....	98
OnSendUserState.....	99
OnTransferPendInfo.....	100
Pickup	101
PrivateState	102
ProxyMakeCall.....	103
ReDial	104
Redirect	105

RemovePhonelcon.....	106
SendCallReminder.....	107
SetAcidReady	108
SetForwarding.....	109
Transfer	110
UnHold	111
UserCallInfo	112
ActiveX	113
Netzwerk	115
NCientActive	115
NlsNetworkVersion.....	116
NLogin	116
NLogoff	116
NNetworkConfigDialog.....	117
NPortNumber	118
NPrepareRestart.....	118
NServerName	119
NStartClient	119
Optionen	120
OAssistedTelephony.....	121
OAutoAnswer.....	121
OAutoConnect.....	122
OAutoLogin	122
OCheckConnection.....	123
OConnectionIconOptions.....	123
ODialKey	124
ODialSpecialKey.....	124
ODropKey	125
ODropSpecialKey.....	125
OHotKey	126
OIniFileName	126
OOptions	127
OOwnerName	128
OOwnerNumber.....	128
OPhoneIcon	128
OShowIcon	129
OShowMenu	129
OUserCallInfo	130
Telefoniefunktionen.....	131
TAddCallToConference.....	132
TAddNumberToConference.....	133
TCallAnswer	133
TCallDrop	134
TCallHold	134
TCallRedirect	134
TCallToggle	135
TCallTransfer	136
TCallUnHold	136
TForwardingDest.....	137
TForwardingState.....	137
TLineReset	137
TMakeCall	138
TPickup	138
TProxyMakeCall.....	139

TReDial	139
TRemovePhoneIcon.....	140
TSendCallReminder.....	140
TSetForwarding.....	141
SQL	142
SQLBegin	143
SQLEnd	143
SQLGetDbBook.....	144
SQLLogin	145
SQLNext	145
Zusatzfunktionen.....	147
EChatEnd	147
EChatMessage.....	148
EChatStart	148
EConvertToCanonical.....	149
EExpandInternNumber.....	150
EGetDateTime	150
EGetFormatDateTime.....	151
ESendUserToUser.....	152
ESetAcidReady.....	152
ESetMainWindow	153
EShowInfoDialog.....	153
EUserSendToDataBase.....	154
Events	155
OnCallNumber	156
OnCallReminder.....	157
OnCallState	158
OnChatMessage.....	158
OnChatStart	159
OnChatStop	159
OnClientClose	160
OnConfNotify	160
OnConnectionState.....	161
OnDatabaseToUser.....	161
OnDbBookRecord.....	162
OnForwardingState.....	163
OnGroupMember.....	163
OnIdentity	164
OnInvalidStringFormat.....	165
OnLicenseInfo	165
OnLocaleInfo	165
OnLogin	166
OnMenuCommand.....	166
OnObjectOfRedirect.....	167
OnProtocolData.....	167
OnRaiseClient	168
OnSendAddress.....	169
OnSendUserState.....	170
OnServerMessage.....	170
OnShowPhoneList.....	171
OnSQLData	171
OnSQLEnd	172
OnSQLLogin	172
OnUserToUser.....	173

C++ CTI API.....	174
Netzwerk	175
CTI_NClientActive.....	176
CTI_NClientRestart.....	176
CTI_NClientStop.....	177
CTI_NGetPortNumber.....	177
CTI_NGetServerName.....	177
CTI_NIsNetworkVersion.....	178
CTI_NLogin	178
CTI_NLogoff	179
CTI_NNetworkConfigDialog.....	179
CTI_NPrepareRestart.....	180
CTI_NSetNetworkOptions.....	181
CTI_NSetPortNumber.....	181
CTI_NSetServerName.....	182
CTI_NStartClientCALLBACK.....	182
CTI_NStartClientHWND.....	183
Optionen	184
CTI_OConnectionIconOptions.....	185
CTI_OGetDialKey.....	186
CTI_OGetDropKey.....	186
CTI_OGetIniFileName.....	187
CTI_OGetOptions.....	188
CTI_OGetOwnerName.....	188
CTI_OGetOwnerNumber.....	189
CTI_OGetUserCallInfo.....	189
CTI_OSetDialKey.....	189
CTI_OSetDropKey.....	190
CTI_OSetOptions.....	191
CTI_OSetOwnerName.....	191
CTI_OSetOwnerNumber.....	192
CTI_OSetUserCallInfo.....	192
Telefoniefunktionen.....	194
CTI_TAddCallToConference.....	195
CTI_TAddNumberToConference.....	196
CTI_TCallAnswer.....	196
CTI_TCallDrop	197
CTI_TCallHold	197
CTI_TCallRedirect.....	198
CTI_TCallToggle.....	198
CTI_TCallTransfer.....	199
CTI_TCallUnHold.....	200
CTI_TGetForwardingDest.....	200
CTI_TGetForwardingState.....	201
CTI_TLineReset.....	201
CTI_TMakeCall	202
CTI_TPickup	202
CTI_TProxyMakeCall.....	203
CTI_TReDial	203
CTI_TRemovePhoneIcon.....	204
CTI_TSendCallReminder.....	204
CTI_TSetForwarding.....	205
SQL	206
CTI_SQLBegin	207

CTI_SQLEnd	207
CTI_SQLGetDbBook.....	208
CTI_SQLLogin	209
CTI_SQLNext	209
Zusatzfunktionen.....	211
CTI_EChatEnd	212
CTI_EChatMessage.....	212
CTI_EChatStart.....	212
CTI_EConvertToCanonical.....	213
CTI_EExpandInternNumber.....	214
CTI_EGetDateTime.....	215
CTI_EGetFormatDateTime.....	215
CTI_ESendUserToUser.....	216
CTI_ESetAcdReady.....	217
CTI_ESetMainWindow	217
CTI_EShow InfoDialog.....	218
CTI_EUserSendToDataBase.....	218
Events	220
WM_CTl_MENUCOMMAND.....	221
WM_CTl_SHOWPHONELIST.....	222
WM_CTl_NETWORKCLOSE.....	223
WM_CTl_NETWORKKOPEN.....	224
WM_CTl_RAISECLIENT.....	225
WM_CTl_NETWORK_DISPATCH.....	226
CTI_CALLNUMBER.....	228
CTI_CALLREMINDER.....	228
CTI_CALLSTATE.....	229
CTI_CHATSTOP.....	230
CTI_CHATMESSAGE.....	230
CTI_CHATSTART.....	231
CTI_CLIENTCLOSE.....	231
CTI_CONFNOTIFY	232
CTI_DATABASESETUSER.....	232
CTI_FORWARDINGSTATE.....	233
CTI_GROUPMEMBER.....	233
CTI_IDENTITY	234
CTI_INVALIDSTRINGFORMAT.....	234
CTI_LICENSEINFO.....	235
CTI_LOCALEINFO.....	235
CTI_LOGIN	236
CTI_OBJECTOFREDIRECT.....	236
CTI_PROTOCOLDATA.....	237
CTI_SENDADDRESS.....	238
CTI_SENDUSERSTATE.....	238
CTI_SERVERMESSAGE.....	239
CTI_SERVICEPAUSE.....	239
CTI_USERTOUSER.....	240
SQL_DATA	240
SQL_DBBOOK.....	241
SQL_END	242
SQL_LOGIN	242
Strukturen	244
CTI_COMMAND_MSG.....	244
CTI_PHONE_DATA.....	245

Datenbankabfragen.....	245
Konstanten	248
Gesprächszustände.....	249
Verbindungsoptionen.....	250
CTICLIENT_ASSISTEDTELEPHONY	250
CTICLIENT_AUTOANSWER.....	251
CTICLIENT_AUTOCONNECT.....	251
CTICLIENT_AUTOLOGIN.....	251
CTICLIENT_CHECKCONNECTION.....	252
CTICLIENT_HOTKEY.....	252
CTICLIENT_PHONEICON.....	252
CTICLIENT_QUICKMOUSEDIAL.....	253
CTICLIENT_SHOWICON.....	253
CTICLIENT_SHOWMENU.....	254
Shortcut Dateien.....	255
Programmierbeispiele	256
ClientCommands.....	257
Einbindung in VB.....	258
Mini CTI Client C++.....	259
Mini CTI Client Delphi.....	260
SQL-Demo.....	261
3 Server-Schnittstelle	263
Befehlsreferenz	264
.NET	265
ClientCom	266
OnUserToDatabase.....	267
SendAdminMessage.....	268
SendIdentity	269
SendUserData	270
CtiConvert	271
ConvertToCanonical.....	272
GetDateTime	273
GetFormatDateTime.....	274
PBXInfo	276
GetExternPrefix.....	277
GetInboundFilter.....	278
GetInternPrefix.....	279
GetNameFromExtension.....	280
GetOutboundFilter.....	281
LicenseCount	282
OnLoginData	283
TMServiceStatus.....	284
Telephony	285
Alternate	286
Answer	287
Conference	288
Forward	289
GetPBXMonitoredLines.....	290
Hangup	291
Init	292
MakeCall	293
OnCallNumber	294
OnCallState	295
OnForwardingState.....	296

OnIdentity	297
OnNew Call	298
OnProtocolData.....	299
Pickup	301
Redirect	302
SendAllUserCallStates.....	303
SendLastCallRecords	304
SendPBXLoginLines.....	305
SetPhoneDisplay.....	306
Transfer	307
ActiveX	308
Funktionen	310
Alternate	311
Answer	312
Conference	312
ConvertToCanonical.....	312
Drop	313
GetAllUserCallStates.....	313
GetDateTime	314
GetExtension	314
GetExternPrefix.....	314
GetFilterCalibrationState.....	315
GetFormatDateTime.....	315
GetInboundFilter.....	316
GetInternPrefix.....	317
GetLastCallRecords.....	318
GetOutboundFilter.....	318
GetPBXFiltersAndPrefixes.....	319
GetPBXMonitoredLines	320
InitCOM	320
LicenseCount	320
MakeCall	321
Pickup	321
Redirect	322
ResetCalibration.....	322
SendIdentity	323
SendUserData	323
SetCalibrationParams.....	324
SetPhoneDisplay.....	324
TMServiceStatus.....	324
Transfer	325
UserCount	326
UserNameFromNumber.....	326
Events	326
OnCallNumber	327
OnCallState	328
OnForwardingState.....	328
OnGetIdentity	329
OnLoginData	330
OnNew Call	330
OnProtocolData.....	331
OnUserToDataBase.....	332
C++ CTI API.....	333
Funktionen	334

CTI_Alternate	334
CTI_Answer	335
CTI_Conference.....	335
CTI_ConvertToCanonical.....	336
CTI_Drop	336
CTI_Forward	337
CTI_GetDateTime.....	337
CTI_GetFormatDateTime.....	338
CTI_GetLicenseCount.....	339
CTI_GetNameFromExtension.....	339
CTI_GetPBXLoginLines	339
CTI_GetPBXMonitoredLines	339
CTI_Init	339
CTI_MakeCall	341
CTI_Pickup	341
CTI_Redirect	342
CTI_SendAdminMessage.....	342
CTI_SendIdentity.....	342
CTI_SendUserData.....	343
CTI_SetPhoneDisplay.....	344
CTI_Transfer	344
Events	345
CTI_CALLNUMBER.....	345
CTI_CALLSTATE.....	346
CTI_FORWARDINGSTATE.....	347
CTI_IDENTITY	347
CTI_NEWCALL.....	348
CTI_PROTOCOLDATA.....	349
CTI_USERTODATABASE.....	349
Programmierbeispiele	351
Anw ahl über IP-Schnittstelle.....	352
ServerCommands.....	353
ServerInterface.....	354
4 Setups	355
Clients ohne Installation verwenden	355
Eingebettetes Client-Setup	356
Eingebettetes Server-Setup	357
Maßgeschneidertes Server-Setup	357
Installationskonfiguration	358
Deinstallation	359
5 Tools	361
CTI-Browser	362
Leitungsbeobachter	365
Setup-Konfigurator	366
Dateien	367
Client-Setup.....	368
Server-Setup.....	369
6 Kundenspezifische Anpassungen	373
Installation	373
Client Schnittstelle	374
CTI Server	375
Serverkonsole	376
Leitungsbeobachter	377

Sprachen	377
Benötigte Ressourcen	378

Index	381
--------------	------------

1 Inhalt

TAPIMaster® SDK Programmierhandbuch

Mit dem TAPIMaster® SDK sind sie in der Lage, innerhalb kurzer Zeit Telefoniefunktionen in Ihre Software einzubauen. TAPIMaster® ersetzt die komplizierte TAPI-Schnittstelle durch einfachere Funktionen, welche sich an den Tätigkeiten des Benutzers orientieren. Anhand von [Beispielen](#) für verschiedene Programmiersprachen finden Sie einen schnellen Einstieg in die praktische Anwendung. Verschiedene [Tools](#) zum [Programmtest](#) und [Installation](#) helfen Ihnen bei der Entwicklung.

[Allgemeines](#)

[Clientschnittstelle](#)

[Serverschnittstelle](#)

[Setups](#)

[Tools](#)

[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.1 Allgemeines

Funktionsumfang

Neben der Telefonie bietet Ihnen TAPIMaster® zahlreiche zusätzliche Möglichkeiten, Ihre Programme mit Funktionen für die Gruppenzusammenarbeit zu bereichern. Für einen Einstieg sind jedoch nur wenige Grundfunktionen notwendig.

Schnittstellen

TAPIMaster® besitzt Schnittstellen auf der [Client](#)- und [Serverseite](#). Sowohl ActiveX als auch normale API sind in jeweils einer DLL vorhanden. Mit ActiveX ist die Programmierung einfacher, mit den API-Funktionen können die Anwendungen ohne Registrierung von Komponenten und ggf. ohne Installation der Clients gestartet werden. TAPIMaster® nimmt Ihnen den Teil der Netzwerkprogrammierung ab, welcher für eine Third Party Telefonie Lösung benötigt wird. In der Einzelplatzversion ist nur eine clientseitige Programmierschnittstelle vorhanden. Der Funktionsumfang ist hier systembedingt eingeschränkt.

Lizenzen

TAPIMaster® erlaubt die Benutzung von bis zu vier des im Programmpaket mitgelieferten Standardclients. Einer dieser fünf Clients kann auch eine Anbindung an ein Fremdprogramm sein. Wenn Sie mehr als einen Client Ihrer eigenen Software mit an die CTI-Schnittstelle anbinden wollen, benötigen Sie eine Lizenz hierfür. Für die Einzelplatzversion ist keine Lizenz erforderlich.

Zeitaufwand

Eine Grundanbindung an die CTI-Schnittstelle ist in wenigen Stunden realisiert. Fast alle [Programmierbeispiele](#) sind das Ergebnis eines Arbeitstags. Nutzen Sie den [CTI-Browser](#) um Funktionen zu testen ohne Code schreiben zu müssen.

Siehe auch

[Inhalt](#) | [Client-Schnittstelle](#) | [Server-Schnittstelle](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2 Client-Schnittstelle

Die Clientschnittstelle besteht im Prinzip aus zwei separaten Schnittstellen in einer DLL. Sie können sowohl ActiveX nutzen als auch auf herkömmliche Art DLL-Funktionen aufrufen. Im zweiten Fall braucht die DLL nicht registriert werden. Testen Sie die Schnittstelle vor Ihrer Verwendung mit dem Standardclient oder dem [CTI-Browser](#).

Die Funktionen und Events haben in beiden Schnittstellen etwa die gleichen Namen. Die Verwendung der wichtigsten Funktionen wird in den verschiedenen [Programmierbeispielen](#) gezeigt.

Die Client-Schnittstellen-DLL gibt es in der Einzelplatz- und der Netzwerkversion. Die Einzelplatzversion ist um einige Funktionen eingeschränkt.

[Befehlsreferenz](#)

[Programmierbeispiele](#)

Siehe auch

[Inhalt](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1 Befehlsreferenz

Die Befehlsreferenz ist in die beiden Schnittstellen für ActiveX und C++API unterteilt. Die C++ Funktionen haben ein CTI_ in den Funktionsnamen vorangestellt.

[.NET](#)
[ActiveX](#)
[C++ CTI API](#)

Siehe auch

[Clientschnittstelle](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1 .NET

Die Client-Schnittstelle besteht aus diesen fünf Klassen. Die Klasse CtiLogin wird zwingend gebraucht, da hier die Initialisierung der Schnittstelle stattfindet.

CtiConvert	Die Klasse dient der Umwandlung von Rufnummern und Datumsformaten.
CtiLogin	Die Klasse enthält Funktionen zur Konfiguration, des Zugriffs zum Client und allgemeine Einstellungen.
Messages	Die Klasse dient dem Senden von Nachrichten von Server zum Client.und umgekehrt.
SQLTool	Die Klasse enthält Befehle und Events zur Abfrage von SQL Daten auf dem Server.
Telephony	Die Klasse enthält alle Funktionen und Events welche mit den Gesprächen direkt zu tun haben.

Siehe auch

[Befehlsreferenz Client](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.1 CtiConvert

Klasse CtiConvert

Die Klasse dient der Umwandlung von Rufnummern und Datumsformaten.

Funktionen

ConvertToCanonical	Umwandlung von Rufnummern in das kanonische Format.
ExpandInternalNumber	Expandiert eine Durchwahl.
GetDateTime	Wandelt time_t in Standardstring um.
GetFormatDateTime	Wandelt time_t in individuell formatierten String um.

Siehe auch

[Übersicht .NET](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.1.1 ConvertToCanonical

Umwandlung von Rufnummern in das kanonische Format.

```
String ConvertToCanonical  
(  
    String strNumber  
)
```

Parameter

strNumber

Rufnummer, die in das kanonische Format umgewandelt werden soll.

Rückgabewert

Umgewandelte Rufnummer

Bemerkungen

Sie können diese Funktion benutzen um Datenbestände mit Rufnummern in das kanonische Format umzuwandeln. Der Ausgabestring sollte mindestens 32 Zeichen groß sein.

Siehe auch

[Übersicht CtiConvert](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.1.2 ExpandInternNumber

Expandiert eine Durchwahl.

```
String ExpandInternNumber  
(  
    String strNumber  
)
```

Parameter

strNumber

Interne Rufnummer, welche erweitert werden soll.

Rückgabewert

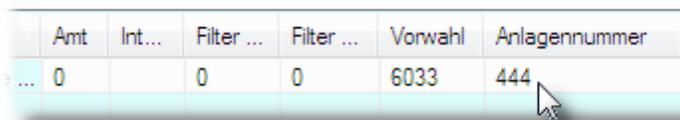
Rückwert, der die expandierte Rufnummer enthält.

Bemerkungen

Erweitert eine Rufnummer um die Landeskenung, Vorwahl und ggf. Anlagenrufnummer.

Beispiel

Bei diesen Einstellungen liefert EExpandInternNumber("22") in Deutschland den Wert +49 (6033) 44422.



Amt	Int...	Filter ...	Filter ...	Vorwahl	Anlagennummer
...	0	0	0	6033	444

Siehe auch

[Übersicht CtiConvert](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.1.3 GetDateTime

Wandelt time_t in Standardstring um.

```
String GetDateTime  
(  
    uint dwTime  
)
```

Parameter

dwTime

Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.

Rückgabewert

Nullterminierter String in einem Standardformat: 2007-01-12T00:53:32.875

Bemerkungen

Nutzen Sie diese Funktion wenn Sie ein gut sortierbares Format brauchen. Viele Programme wie Excel nutzen dieses Format.

Siehe auch

[Übersicht CtiConvert](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.1.4 GetFormatDateTime

Wandelt time_t in individuell formatierten String um.

```
String GetFormatDateTime
(
    uint dwTime,
    String strFormat
)
```

Parameter

lTime

Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.

strFormat

String mit Datumsformat. Dieses kann aus den folgenden Formatzeichen gebildet werden.

Zeichen	Erklärung
%a	Abgekürzter Wochentag
%A	Ausgeschriebener Wochentag
%b	Abgekürzter Monatsname
%B	Ausgeschriebener Monatsname
%c	Lokales Datum und Uhrzeit, abhängig von Systemeinstellungen
%d	Monatstag (01 – 31)
%H	Stunde im 24Stunden-Format (00 – 23)
%I	Stunde im 12Stunden-Format (01 – 12)
%j	Tag im Jahr (001 – 366)
%m	Monat als Zahl (01 – 12)
%M	Minute (00 – 59)
%p	Lokaler Indikator für vormittag / nachmittag bei 12stündiger Anzeige
%S	Sekunden (00 – 59)
%U	Kalenderwoche, beginnend mit Sonntag (00 – 53)
%w	Wochentag als Zahl, beginnend mit Sonntag (0 – 6)
%W	Kalenderwoche, beginnend mit Montag (00 – 53)
%x	Lokales Datum, abhängig vom System
%X	Lokale Zeit, abhängig vom System
%y	2stellige Jahresangabe (00 – 99)
%Y	4stellige Jahresangabe
%z,%Z	Zeitzone
%%	Prozentzeichen

Rückgabewert

String in einem individuellen Format.

Bemerkungen

Die Formatzeichen können beliebig kombiniert werden.

Siehe auch

[Übersicht CtiConvert](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2 CtiLogin

Klasse CtiLogin

Die Klasse enthält Funktionen zur Konfiguration, des Zugriffs zum Client und allgemeine Einstellungen.

Funktionen

AssistedTelephony	Unterstützung von Assisted Telephony zuschalten.
AutoAnswer	Gespräche werden automatisch angenommen.
AutoConnect	Die Netzwerkverbindung wird selbständig gehalten.
AutoLogin	Selbständig an CTI-Server anmelden.
BeforeClose	Wird aufgerufen bevor das Programm beendet wird.
CheckConnection	Die Verbindungsdaten werden beim Start geprüft.
ClientActive	Schaltet Netzwerkverbindung zu oder ab.
ConfigFileName	Der Name der Konfigurationsdatei wird zurückgeliefert.
HotKey	Schaltet die Schnellwahl über Tastaturcodes zu.
Login	Meldet Client am CTI-Server an.
Logoff	Meldet Client vom CTI-Server ab.
NetworkConfigDialog	Zeigt Netzwerkdialog an.
OnClientClose	Der Client soll geschlossen und auf den Neustart vorbereitet werden.
OnConnectionState	Der Zustand der Verbindung hat sich geändert.
OnLicenseInfo	Liefert erweiterte Lizenzinformationen.
OnLocaleInfo	Liefert Anlagenparameter und Landeseinstellungen.
OnLogin	Die Anmeldung am CTI-Server wird bestätigt.
OnMenuCommand	Das Menü des Verbindungssicons wurde angeklickt.
OnRaiseClient	Teilt dem Client mit dass er sich in den Vordergrund stellen soll.
OnShowPhoneList	Der Benutzer hat einen Doppelklick auf das gelbe Telefonsymbol getätigt.
OwnName	Der Anmeldenname wird festgelegt.
OwnNumber	Die eigene Durchwahl wird festgelegt.
Phonelcon	Ein Telefonsymbol wird bei eingehenden Anrufen eingeblendet.
Port	Ändert Port der Netzwerkverbindung.
PrepareRestart	Bereitet Client auf Neustart vor.
QuickMouseDial	Zuschaltung der Schnellwahl über Kontextmenü.
SaveSettings	Wird aufgerufen wenn die Schnittstelle ihre Einstellungen speichern soll.
ServerName	Ändert den Servernamen der Netzwerkverbindung.

SetConnectionIconOptions	Das Aussehen des Verbindungsicons wird geändert.
SetMainWindow	Übergibt das Hauptfensterhandle an die Schnittstelle.
ShowIcon	Ein Verbindungssymbol wird angezeigt.
ShowInfoDialog	Zeigt Versionsinfo der Schnittstelle an.
ShowMenu	Das Verbindungsicon wird mit einem Menü versehen.
StartClient	Initialisiert der CTI-Schnittstelle.

Siehe auch[Übersicht .NET](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.1 AssistedTelephony

Unterstützung von Assisted Telephony zuschalten.

Eigenschaft AssistedTelephony

Typ

bool, lesen und schreiben

Bemerkungen

Nimmt Befehle für Assisted Telephony Funktionen entgegen. Wenn aus einem Fremdprogramm die Windows TAPI Funktion `tapiRequestMakeCall` aufgerufen wird, so startet die Schnittstelle einen Anruf.

Voreinstellung

true

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.2 AutoAnswer

Gespräche werden automatisch angenommen.

Eigenschaft AutoAnswer**Typ**

bool, lesen und schreiben

Bemerkungen

Eingehende Gespräche werden automatisch angenommen. Diese Option macht Sinn in Callcentern. Das Telefon / Headset muss ins Freisprechen schalten können und der TAPI Treiber dies unterstützen.

Voreinstellung

false

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.3 AutoConnect

Die Netzwerkverbindung wird selbständig gehalten.

Eigenschaft AutoConnect

Typ

bool, lesen und schreiben

Bemerkungen

Beim Programmstart stellt die CTI-Schnittstelle selbständig die Verbindung zum CTI-Server her. Wird diese Verbindung unterbrochen, so versucht der Client diese wieder aufzubauen. [ClientActive](#) muss nicht extra auf True gesetzt werden.

Voreinstellung

true

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.4 AutoLogin

Selbständig an CTI-Server anmelden.

Eigenschaft AutoLogin**Typ**

bool, lesen und schreiben

Bemerkungen

Die Schnittstelle meldet sich nach Aufbau der Netzwerkverbindung selbständig am Server an, [Login](#) muss nicht separat aufgerufen werden.

Voreinstellung

true

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.5 BeforeClose

Wird aufgerufen bevor das Programm beendet wird.

void BeforeClose()

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Die Funktion räumt die Client Schnittstelle intern auf.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.6 CheckConnection

Die Verbindungsdaten werden beim Start geprüft.

Eigenschaft CheckConnection**Typ**

bool, lesen und schreiben

Bemerkungen

Beim Programmstart prüft die Schnittstelle die Verbindungsparameter auf Vollständigkeit. Das wären Servername, Portnummer und interne Durchwahl. Fehlt ein Parameter, so bringt die Schnittstelle den [Anmeldedialog](#) nach vorne.

Voreinstellung

true

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.7 ClientActive

Schaltet Netzwerkverbindung zu oder ab.

Eigenschaft ClientActive

Typ

bool, lesen und schreiben

Bemerkungen

Setzen Sie diese Eigenschaft auf False wenn Sie eine bestehende Netzwerkverbindung zum CTI-Server beenden möchten. Ein Wert von True stellt die Verbindung wieder her.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.8 ConfigFileName

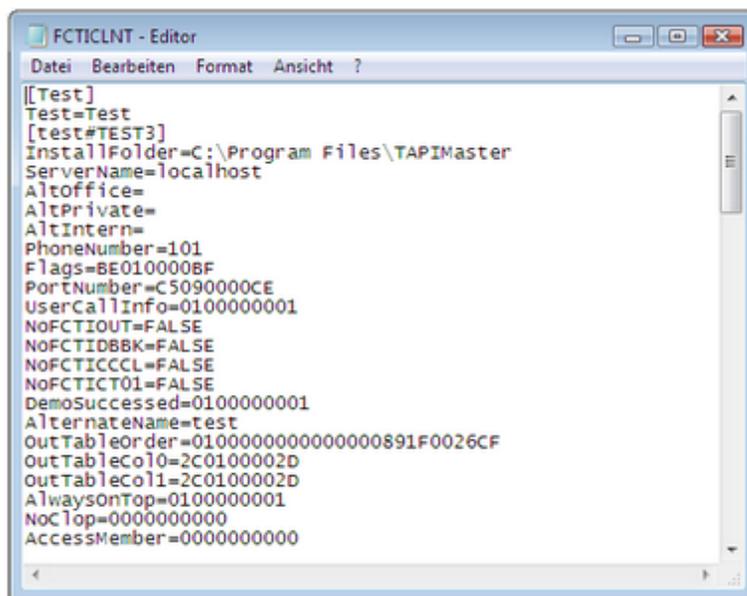
Der Name der Konfigurationsdatei wird zurückgeliefert.

Eigenschaft ConfigFileName**Typ**

String, nur lesen

Bemerkungen

Die CTI-Schnittstelle speichert ihre Einstellungen in einer Konfigurationsdatei. Wenn Sie als Programmierer für die Einstellungen Ihres Clients keine separate Datei anlegen wollen, können Sie diese verwenden. Die Datei liegt im Windows-INI Format vor. Der Sektionsname ergibt sich aus dem Benutzernamen + # + Computernamen. Beispiel: [mueller#wk12]. Werfen Sie einen Kurzen Blick in eine der vorhandenen INI-Dateien, um keine vorhandenen Schlüsselwörter zu benutzen.



```
[Test]
Test=Test
[test#TEST3]
InstallFolder=C:\Program Files\TAPIMaster
ServerName=localhost
AltOffice=
AltPrivate=
AltIntern=
PhoneNumber=101
Flags=BE01000BF
PortNumber=C509000CE
UserCallInfo=0100000001
NoFCTIOUT=FALSE
NoFCTIDBBK=FALSE
NoFCTICCL=FALSE
NoFCTICT01=FALSE
DemoSucceeded=0100000001
AlternateName=test
OutTableOrder=010000000000000891F0026CF
OutTableCol0=2C0100002D
OutTableCol1=2C0100002D
AlwaysOnTop=0100000001
NoClop=0000000000
AccessMember=0000000000
```

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.9 HotKey

Schaltet die Schnellwahl über Tastaturcodes zu.

Eigenschaft HotKey

Typ

bool, lesen und schreiben

Bemerkungen

Legt fest, ob markierter Text mit einer Tastenkombination angewählt werden kann. Das kann eine der Tasten F1 bis F12 sein, Zusätzlich können die Umschalt- und / oder die Strg-Taste gedrückt sein. Befindet sich eine wählbare Nummer im Text, so wird diese angewählt. Die Schnellwahl- und Zusatz Tasten werden mit [DialKey](#) und [DialSpecialKey](#) festgelegt.

Voreinstellung

false

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.10 Login

Meldet Client am CTI-Server an.

void Login()**Parameter**

Keine

Rückgabewert

keiner

Bemerkungen

Die Funktion wird automatisch ausgeführt, wenn [AutoLogin](#) aktiv ist und die Netzwerkverbindung steht. **Login** schaltet im CTI-Server die Funktionalität für den Client frei. Die anderen Gruppenmitglieder sehen den Client dann als angemeldet. Wenn Sie die Schnittstelle in ein Programm einbinden was den ganzen Tag über läuft kann es sinnvoll sein die CTI-Funktionalität zeitweilig abzuschalten, z. B. während der Mittagspause. Die anderen Gruppenmitglieder sehen dann dass der Benutzer nicht ansprechbar ist. [Logoff](#) schaltet die CTI ab und **Login** wieder an.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.11 Logoff

Meldet Client vom CTI-Server ab.

void Logoff()

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Schaltet die CTI-Funktionalität im CTI-Server ab. Die Gruppenmitglieder sehen den Client dann als abgemeldet. Alternativ dazu kann auch die Netzwerkverbindung komplett mit gekappt werden wenn [ClientActive](#) auf False gesetzt wird.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.12 NetworkConfigDialog

Zeigt Netzwerkdialog an.

void NetworkConfigDialog()**Parameter**

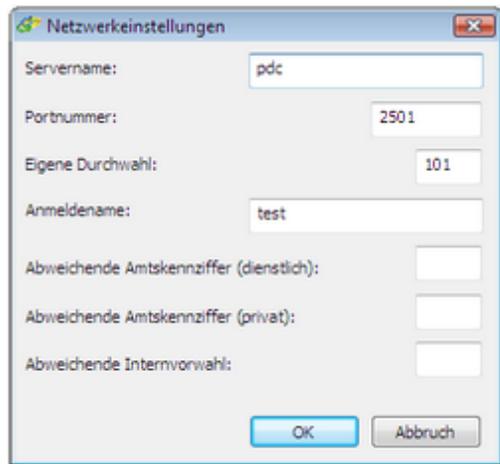
Keine

Rückgabewert

keiner

Bemerkungen

Ein Dialog wird geöffnet in welchem die Verbindungsdaten eingestellt werden können. Alternativ dazu ist die Verwendung von [ServerName](#), [Port](#) und [OwnNumber](#) möglich. Fehlen die ersten drei Angaben und ist die Eigenschaft [CheckConnection](#) auf True gesetzt, so erscheint der Dialog automatisch beim Programmstart. Für die Anzeige des Dialogs wird die Verbindung zum CTI-Server unterbrochen. Mit OK wird eine neue Verbindung hergestellt. Benutzen Sie diesen Dialog wenn Sie nach einer Möglichkeit suchen die Verbindungsparameter zu ändern und selbst keine Eingabemöglichkeit programmieren wollen.

**Siehe auch**

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.13 OnClientClose

Der Client soll geschlossen und auf den Neustart vorbereitet werden.

```
Event OnClientClose  
(  
    object sender,  
    int e.nMinutes  
)
```

Parameter

sender

Objekt [CtiLogin](#)

e.nMinutes

Zeitspanne in Minuten, nach deren Verlauf der Client neu gestartet werden soll, wenn der Wert positiv oder 0.

Bemerkungen

Diese Nachricht schickt der Administrator, wenn er wichtige Dateien austauschen will. Das macht Sinn wenn die Clients aus einem freigegebenen Netzwerkverzeichnis gestartet wurden. Dazu müssen alle Clients heruntergefahren werden. Wenn Sie dieses Feature unterstützen möchten rufen Sie hier [PrepareRestart](#) auf und übergeben *IMinutes* als Parameter. Bei einem negativen Wert von *nMinutes* soll die Anwendung nicht neu gestartet werden. Schließen Sie danach Ihre Anwendung. **OnClientClose** wird über die Serverkonsole ausgelöst (Seite Benutzerverwaltung).

**Siehe auch**

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.14 OnConnectionState

Der Zustand der Verbindung hat sich geändert.

Event OnConnectionState

```
(  
    object sender,  
    int e.nState  
)
```

Parameter

sender

Objekt [CtiLogin](#)

e.nState

Zustand, in dem sich die Verbindung zum Server befindet.

Bemerkungen

Der Zustand der Netzwerkverbindung hat sich geändert.

Mögliche Werte für den Zustand sind:

0: Unbekannt

1: Keine Netzwerkverbindung

2: Verbindung steht, aber Server arbeitet nicht.

3: Verbindung steht, CTI-Server arbeitet.

Der Verbindungszustand ändert sich beim Verbinden des Clients mit dem Server oder wenn der Server heruntergefahren wird bzw. der Service eine Pause einlegt.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.15 OnLicenseInfo

Liefert erweiterte Lizenzinformationen.

```
Event OnLicenseInfo  
(  
    object sender,  
    String e.strMessage  
)
```

Parameter

sender

Objekt [CtiLogin](#)

e.strMessage

Erweiterte Lizenzinformationen. Die übermittelten Werte sind von geschweiften Klammern eingerahmt und durch Semikolons getrennt. Formatbeispiel:
{License for: Musterfirma;Street: Musterstraße 1;City: 12345 Königs-Musterhausen;Phone: +49 (5555) 664488;Web: http://www.musterfirma.de}

Bemerkungen

Die Nachricht wird nach dem Anmelden gesendet.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.16 OnLocaleInfo

Liefert Anlagenparameter und Landeseinstellungen.

```
Event OnLocaleInfo  
(  
    object sender,  
    int e.nInternLength,  
    inte. nCountry,  
    String e.strCity,  
    String e.strPBX  
)
```

Parameter

sender

Objekt [CtiLogin](#)

e.nInternLength

Maximale Länge interner Durchwahlen, Werte von 2 bis 4 sind möglich.

e.nCountry

Landeskennung, z. B. 49 für Deutschland

e.strCity

Vorwahl ohne führende 0

e.strPBX

Anlagenrufnummer. Wird bei Telefonanlagen mit Amtsanschluss verwendet. Bei einer Rufnummer +49 (69) 123456-789 wäre 123456 die Anlagenrufnummer.

Bemerkungen

Event liefert bei jeder Anmeldung diese lokalen Parameter zurück, die dann im Programm weiter verwendet werden können.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.17 OnLogin

Die Anmeldung am CTI-Server wird bestätigt.

```
Event OnLogin  
(  
    object sender,  
    bool e.bSuccess  
)
```

Parameter

sender

Objekt [CtiLogin](#)

e.bSuccess

Erfolgsstatus der Anmeldung, ist True wenn der Benutzer sich anmelden konnte.

Bemerkungen

Eine erfolgreiche Anmeldung ist die Voraussetzung dafür dass die meisten Befehle abgesetzt werden können und Events empfangen werden.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.18 OnMenuCommand

Das Menü des Verbindungsicons wurde angeklickt.

```
Event OnMenuCommand  
(  
    object sender,  
    uint e.dwCommand  
)
```

Parameter

sender

Objekt [CtiLogin](#)

e.dwCommand

Befehls-ID

Bemerkungen

Der Benutzer hat auf das Menü des Verbindungssymbols geklickt. Der ausgelöste Befehl wird in *dwCommand* übertragen.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.19 OnRaiseClient

Teilt dem Client mit dass er sich in den Vordergrund stellen soll.

```
Event OnRaiseClient  
(  
    object sender  
)
```

Parameter

sender

Objekt [CtiLogin](#)

Bemerkungen

Ein Doppelklick auf das Netzwerksymbol löst dieses Event aus. Der Benutzer möchte die Telefonieanwendung im Vordergrund sehen.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.20 OnShow PhoneList

Der Benutzer hat einen Doppelklick auf das gelbe Telefonsymbol getätigt.

Event OnShowPhoneList

```
(  
    object sender  
)
```

Parameter

sender

Objekt [CtiLogin](#)

Bemerkungen

Bei eingehenden Anrufen kann ein gelbes Telefonsymbol in der Taskleiste angezeigt werden. Ein Doppelklick darauf löst diesen Event aus. Das Programm sollte dann daraufhin die Anruferliste anzeigen, falls es eine solche Liste gibt.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.21 OwnName

Der Anmeldename wird festgelegt.

Eigenschaft OwnName

Typ

String, lesen und schreiben

Bemerkungen

Die Änderung bewirkt eine Neuansmeldung am Server.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.22 OwnNumber

Die eigene Durchwahl wird festgelegt.

Eigenschaft OwnNumber**Typ**

String, lesen und schreiben

Bemerkungen

Eigene Durchwahl, welcher der Server benötigt um die Leitung zuzuordnen.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.23 Phonelcon

Ein Telefonsymbol wird bei eingehenden Anrufen eingeblendet.

Eigenschaft Phonelcon

Typ

bool, lesen und schreiben

Bemerkungen

Bei eingehenden Anrufen kann in der Taskleiste ein kleines Telefonsymbol eingeblendet werden, analog dem Briefsymbol in MS Outlook. Im Tooltip des Icons wird dann die Rufnummer sofern vorhanden angezeigt. Das Icon verfügt über ein kleines Menü, mit dem der Anrufer zurückgerufen werden kann. Das Icon wird mit [RemovePhonelcon](#) entfernt. Ein Doppelklick auf das Icon löst den Event [OnShowPhoneList](#) aus, hier sollten Sie die Anruferliste sofern vorhanden anzeigen.



Voreinstellung

true

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.24 Port

Ändert Port der Netzwerkverbindung.

Eigenschaft Port**Typ**

int32, lesen und schreiben

Bemerkungen

Die Portnummer der Verbindung. Wird der Wert geändert, so gibt es keinen Neuaufbau der Verbindung mit den geänderten Parametern. Um eine neue Netzwerkverbindung zu starten schalten Sie die Option [ClientActive](#) aus und wieder an.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.25 PrepareRestart

Bereitet Client auf Neustart vor.

```
void PrepareRestart  
(  
    int nMinutes  
)
```

Parameter

nMinutes

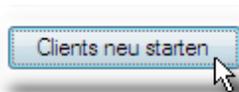
Zeit in Minuten, die bis zum Neustart vergehen soll. Bei *nMinutes* = 0 erfolgt der Neustart sofort, bei negativen Werten wird der Client nicht neu gestartet.

Rückgabewert

keiner

Bemerkungen

Der CTI-Server kann die Standardclients herunterfahren und wieder neu starten. Das kann sinnvoll sein wenn die Clients aus einem Netzwerk heraus gestartet wurden und nun die Dateien ausgetauscht werden sollen. Über die Serverkonsole auf der Seite "Benutzerverwaltung" kann der Neustart angestoßen werden.



Sie können diese Funktionen ebenfalls unterstützen. Wenn ein Event [OnClientClose](#) eintritt, so bekommen Sie in dessen *nMinutes* die Zeit mitgeteilt, in welcher der Client neu starten soll. Rufen Sie jetzt [PrepareRestart](#) mit dieser Zeit als Parameter auf und beenden Sie die Anwendung. Sie wird nach der verstrichenen Zeit neu gestartet.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.26 QuickMouseDial

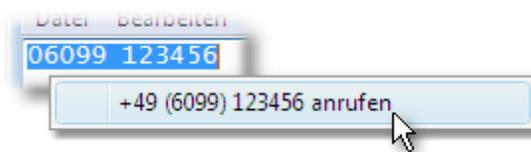
Zuschaltung der Schnellwahl über Kontextmenü.

Eigenschaft QuickMouseDial**Typ**

bool, lesen und schreiben

Bemerkungen

Mit Hilfe dieser Option kann markierter Text über ein Kontextmenü angewählt werden. Befindet sich eine wählbare Nummer im Text, so blendet sich ein solches Kontextmenü ein:

**Voreinstellung**

false

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.27 SaveSettings

Wird aufgerufen wenn die Schnittstelle ihre Einstellungen speichern soll.

void SaveSettings()

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Normalerweise werden die Einstellungen beim Programmende gespeichert. Mit dieser Funktion kann man es zwischendurch tun. Die Einstellungen werden in einer Datei geschrieben, die mit [ConfigFileName](#) abfragbar ist.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.28 ServerName

Ändert den Servernamen der Netzwerkverbindung.

Eigenschaft ServerName**Typ**

String, lesen und schreiben

Bemerkungen

Der Name des CTI-Servers. Das kann auch die IP-Adresse oder "localhost" sein. Wird der Wert geändert, so gibt es keinen Neuaufbau der Verbindung mit den geänderten Parametern. Um eine neue Netzwerkverbindung zu starten schalten Sie die Option [ClientActive](#) aus und wieder an.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.29 SetConnectionIconOptions

Das Aussehen des Verbindungsicons wird geändert.

```
void ConnectionIconOptions  
(  
    uint hlconOn,  
    uint hlconOff,  
    uint hMenu,  
    String strText  
);
```

Parameter*hlconOn*

Handle des Icons, welches bei bestehender Netzwerkverbindung zum CTI Server angezeigt wird. Bei 0 wird das Standardicon angezeigt.

hlconOff

Handle des Icons, welches bei unterbrochener Netzwerkverbindung zum CTI Server angezeigt wird. Bei 0 wird das Standardicon angezeigt.

hMenu

Handle auf ein Menü, welches über dem Verbindungsicon angezeigt wird, wenn dieser mit der rechten Maustaste darauf klickt. Kann 0 sein, dann wird kein Menü verwendet.

strText

Tooltip, welcher über dem Verbindungsicon angezeigt wird. Maximal 63 Zeichen. Kann Null sein, dann wird ein Standardtext angezeigt. Soll kein Tooltip angezeigt werden, so übergeben Sie einen Leerstring.

Rückgabewert

keiner

Bemerkungen

Aussehen, Menü und Tooltips des Verbindungsicons lassen sich ändern. Für den Tooltiptext gibt es ebenfalls einen Standardtext. Um das Verbindungsicon nutzen zu können, muss die Option [ShowIcon](#) aktiviert sein. Bei Menüs ist zusätzlich [ShowMenu](#) erforderlich.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.30 SetMainWindow

Übergibt das Hauptfensterhandle an die Schnittstelle.

```
void SetMainWindow  
(  
    uint hWnd  
)
```

Parameter

hWnd
Fensterhandle der Anwendung

Rückgabewert

keiner

Bemerkungen

Übergibt der Schnittstelle das Handle des Hauptfensters der Anwendung. Wird benötigt, um Unterdialoge der Schnittstelle an der richtigen Position anzuzeigen.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.31 Show Icon

Ein Verbindungssymbol wird angezeigt.

Eigenschaft ShowIcon**Typ**

bool, lesen und schreiben

Bemerkungen

In der Taskleiste wird ein Verbindungssymbol angezeigt, welches den Status der Netzwerkverbindung erkennen lässt. Es kann mit einem benutzerdefinierten Menü und Tooltip versehen werden. Die Icons für stehende unterbrochene Verbindung können durch eigene Icons ausgetauscht werden.

**Voreinstellung**

true

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.32 Show InfoDialog

Zeigt Versionsinfo der Schnittstelle an.

void ShowInfoDialog()

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Zeigt einen kleinen Dialog mit Informationen zur Schnittstelle an. Hersteller, Produkt- und Dateiversion der Schnittstelle werden angezeigt.

**Siehe auch**

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.33 Show Menu

Das Verbindungsicon wird mit einem Menü versehen.

Eigenschaft ShowMenu

Typ

bool, lesen und schreiben

Bemerkungen

Dem Verbindungsicon in der Taskleiste lässt sich ein Menü zuordnen. Es erscheint dann wenn der Benutzer mit der rechten Maustaste auf das Icon klickt. Mit [SetConnectionIconOptions](#) wird das Menühandle übergeben. Die Events des Menüs werden mit OnMenuCommand geliefert. [ShowIcon](#) muss True sein.

Voreinstellung

false

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.2.34 StartClient

Initialisiert der CTI-Schnittstelle.

```
void StartClient()
```

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Mit dieser Funktion wird die [Schnittstelle](#) initialisiert. Ab nun können Events empfangen werden.

Siehe auch

[Übersicht CtiLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3 Messages

Klasse Messages

Die Klasse dient dem Senden von Nachrichten von Server zum Client.und umgekehrt.

Funktionen

ChatEnd	Die aktive Chatsitzung wird beendet.
ChatMessage	Die Chatnachricht wird übertragen.
ChatStart	Startet eine Chatsitzung.
OnChatMessage	Eine Chatnachricht wird vom Chatpartner gesendet.
OnChatStart	Eine Chat-Sitzung wurde eröffnet.
OnChatStop	Die Chatsitzung soll beendet werden.
OnDatabaseToUser	Eine benutzerdefinierte Nachricht wurde vom Server geschickt.
OnServerMessage	Der Administrator schickt dem Benutzer eine Nachricht.
OnUserToUser	Ein anderer Benutzer schickt eine Nachricht.
SendUserToUser	Eine Kurznachricht wird geschickt.
UserSendToDatabase	Sendet benutzerdefinierte Nachrichten zur Datenbankschnittstelle.

Siehe auch

[Übersicht .NET](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.1 ChatEnd

Die aktive Chatsitzung wird beendet.

void ChatEnd()

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Jeder der beiden Chatpartner kann diese Funktion aufrufen. Darauf hin erhalten beide Seiten einen Event vom Typ [OnChatStop](#). Die Anwendung kann dann die Chatfenster schließen oder deaktivieren.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.2 ChatMessage

Die Chatnachricht wird übertragen.

```
void ChatMessage  
(  
    String strMessage  
)
```

Parameter

strMessage
Nachrichtentext, der in der Chatbox steht.

Rückgabewert

keiner

Bemerkungen

Es wird der komplette Text in der Chatbox übertragen. Die Funktion sollte aufgerufen werden sobald sich im jeweiligen Eingabefeld etwas ändert. Die Funktion ruft beim Chatpartner den Event [OnChatMessage](#) auf.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.3 ChatStart

Startet eine Chatsitzung.

```
void ChatStart  
(  
    String strDest  
)
```

Parameter

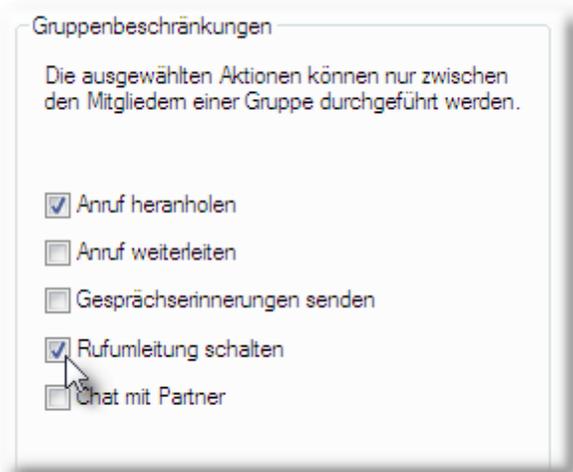
strDest
Durchwahl des Chatpartners

Rückgabewert

keiner

Bemerkungen

Der Chatpartner muss angemeldet und die Rechte zum Chat vorhanden sein; es darf seinerseits noch keine Chatsitzung geben. Wenn die Anmeldung zum Chat zulässig war, erhalten beide Partner einen Event vom Typ [OnChatStart](#). Daraufhin kann die Anwendung Ein- und Ausgabefelder für die Chatsitzung einblenden. Sollte Chat nicht funktionieren könnte es daran liegen dass Chat auf die Mitglieder der gleichen Gruppe beschränkt ist. Auf der Seite "Gruppenverwaltung" der Serverkonsole können Sie diese Beschränkungen aufheben.

**Siehe auch**

[Übersicht Messages](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.4 OnChatMessage

Eine Chatnachricht wird vom Chatpartner gesendet.

```
Event OnChatMessage  
(  
    object sender,  
    String e.strMessage  
)
```

Parameter

sender

Objekt [Messages](#)

e.strMessage

Nachricht, maximal 255 Zeichen

Bemerkungen

Ein Chat mit einem anderen Teilnehmer besteht und dieser hat uns eine Nachricht geschickt. Zeigen Sie diese Nachricht im Ausgabefenster für diese Nachrichten an. Es wird nicht nur der letzte Buchstabe, sondern der gesamte Text übertragen.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.5 OnChatStart

Eine Chat-Sitzung wurde eröffnet.

```
Event OnChatStart  
(  
    object sender,  
    bool e.bStartOK,  
    String e.strUser  
)
```

Parameter

sender

Objekt [Messages](#)

e.bStartOK

Zeigt an ob die Sitzung erfolgreich eröffnet wurde.

e.strUser

Durchwahl des anderen Chatpartners

Bemerkungen

Es Gibt zwei Möglichkeiten, dieses Event zu erhalten:

1. Sie haben mit [ChatStart](#) um eine Chatsitzung mit einem Teilnehmer gebeten. Diese kann von den Rechten her nicht erlaubt sein oder der Chatpartner hat bereits eine Sitzung. Dann ist *bStartOK* False. Ansonsten bekommen Sie True zurück, in dem Fall können Sie geeignete Fenster für die Chat Ein- und Ausgabe anzeigen.
2. Jemand möchte eine Chatsitzung mit Ihnen beginnen. Anhand von *bstrUser* sehen Sie wer das ist. Wenn Sie die Chatsitzung ablehnen möchten, so rufen Sie hier [ChatEnd](#) auf. Ansonsten zeigen Sie die Fenster für Ein- und Ausgabe an.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.6 OnChatStop

Die Chatsitzung soll beendet werden.

```
Event OnChatStop  
(  
    object sender  
)
```

Parameter

sender

Objekt [Messages](#)

Bemerkungen

Die Nachricht wird an beide Chatpartner gesendet, wenn einer von ihnen die Chatsitzung schließt.
Die Chatfenster sollten daraufhin geschlossen werden.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.7 OnDatabaseToUser

Eine benutzerdefinierte Nachricht wurde vom Server geschickt.

```
Event OnDatabaseToUser  
(  
    object sender,  
    uint e.dwCommand,  
    String e.strMessage  
)
```

Parameter

sender

Objekt [Messages](#)

e.dwCommand

Typ des benutzerdefinierten Kommandos

e.strMessage

Übermittelte Parameter

Bemerkungen

TAPIMaster® besitzt auf der Serverseite eine zweite Schnittstelle. Normalerweise werden Datenbankabfragen über die ODBC-Schnittstelle des Servers getätigt. Alternativ dazu kann die Serverschnittstelle jedoch in einer Datenbank oder anderen Anwendung eingebunden werden. Über [UserSendToDataBase](#) können dann benutzerdefinierte Daten getauscht werden.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.8 OnServerMessage

Der Administrator schickt dem Benutzer eine Nachricht.

```
Event OnServerMessage  
(  
    object sender,  
    String e.strMessage  
)
```

Parameter

sender

Objekt [Messages](#)

e.strMessage

Nachrichtentext

Bemerkungen

Der Administrator schickt eine Nachricht an den Benutzer, welche in *bstrMessage* enthalten ist. Es werden bis max. 255 Zeichen gesendet. Die Nachrichten werden von der Serverkonsole auf der Seite "Benutzerverwaltung" gesendet.

**Siehe auch**

[Übersicht Messages](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.9 OnUserToUser

Ein anderer Benutzer schickt eine Nachricht.

```
Event OnUserToUser
(  
    object sender,  
    String e.strNumber,  
    String e.strMessage  
)
```

Parameter

sender

Objekt [Messages](#)

e.strNumber

Durchwahl des anderen Teilnehmers

e.strMessage

Textnachricht, die er verschickt, maximal 255 Zeichen.

Bemerkungen

Die Benutzer können sich untereinander mit [SendUserToUser](#) Nachrichten zukommen lassen. Wenn Sie diese Funktion implementieren können Sie möglicherweise an manchen Arbeitsplätzen auf einen Mailclient verzichten.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.10 SendUserToUser

Eine Kurznachricht wird geschickt.

```
void SendUserToUser  
(  
    String strOtherUser,  
    String strMessage  
)
```

Parameter

strOtherUser

Durchwahl des anderen Teilnehmers

strMessage

Kurznachricht (max 220 Zeichen), die an ihn geschickt wird.

Rückgabewert

keiner

Bemerkungen

Verschiedene Benutzer können sich mit dieser Funktion Kurznachrichten schicken, ohne den Mailclient bemühen zu müssen. Auf dem Client des Empfängers wird der Event [OnUserToUser](#) ausgelöst.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.3.11 UserSendToDataBase

Sendet benutzerdefinierte Nachrichten zur Datenbankschnittstelle.

```
void UserSendToDataBase  
(  
    uint dwCommand,  
    String strText  
)
```

Parameter

dwCommand
Nummer des Befehls

strText
Übermittelte Parameter

Rückgabewert

keiner

Bemerkungen

Mit dieser Funktion können benutzerdefinierte Nachrichten an die Serverschnittstelle geschickt werden. Dort lösen diese einen Event vom Typ [OnUserToDataBase](#) aus. Anhand von *ICommand* werden die dort eingehenden Befehle unterschieden. Die Serverschnittstelle kann auf die Nachricht antworten und löst auf der Clientseite den Event [OnDatabaseToUser](#) aus.

Siehe auch

[Übersicht Messages](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4 SQLTool

Klasse SQLTool

Die Klasse enthält Befehle und Events zur Abfrage von SQL Daten auf dem Server.

Funktionen

Begin	Eine SQL-Abfrage wird gestartet.
End	Eine SQL-Abfrage wird beendet.
GetDbBook	Rufnummerdatensätze zu Namen oder Namensbestandteilen werden gesucht.
Login	Die Anmeldung an die Datenbank erfolgt.
Next	SQL-Abfrage wird fortgesetzt.
OnDbBookRecord	Ein Datensatz aus der Rufnummernsuche wird übermittelt.
OnSQLData	SQL-Abfragen werden beantwortet.
OnSQLEnd	Eine SQL-Abfrage wurde beendet.
OnSQLLogin	Die Anmeldung an die Datenbank wird bestätigt.

Siehe auch

[Übersicht .NET](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.1 Begin

Eine SQL-Abfrage wird gestartet.

```
uint Begin
(String strCommand,
uint dwCount)
```

Parameter

strCommand

SQL-Kommando, welches an die Datenbank geschickt wird.

dwCount

Maximale Anzahl der Datensätze, die abgefragt werden soll.

Rückgabewert

Abfrage-ID als Rückgabewert

Bemerkungen

Die Funktion startet eine Datenbankabfrage. Es kann ein beliebiger SQL-String übergeben werden, sofern der Administrator hier keine Beschränkungen vorgenommen hat. Da die Zahl der ermittelten Datensätze sehr groß werden kann sollte sie mit *dwCount* sinnvoll beschränkt werden. Die Abfrage

lässt sich mit [Next](#) fortsetzen und [End](#) beenden. Mit **Begin** gestartete Abfragen werden nach einigen Minuten selbständig auf dem Server terminiert. Bei jeder Abfrage wird eine Abfrage-ID zurückgeliefert. Diese wird beim Aufruf von [Next](#) und [End](#) benötigt. Wird 0 zurückgeliefert so konnte die Abfrage nicht gestartet werden. Das ist der Fall wenn der Benutzer nicht an der Datenbank angemeldet ist.

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.2 End

Eine SQL-Abfrage wird beendet.

```
void End  
(  
    uint dwRequest  
)
```

Parameter

dwRequest
Abfrage-ID

Rückgabewert

keiner

Bemerkungen

Eine Abfrage wird beendet. Daraufhin erhält der Client einen Event vom Typ [OnSQLEnd](#). Da für jede Abfrage ein Statementhandle im Datenbankzugriff angelegt wird, sollten nicht benötigte Abfragen so schnell wie möglich terminiert werden. Beim Aufruf von **End** übergeben Sie die Abfrage-ID, welche bei [Begin](#) zurückgeliefert wurde. Nicht geschlossene Abfragen werden nach einigen Minuten automatisch beendet.

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.3 GetDbBook

Rufnummerdatensätze zu Namen oder Namensbestandteilen werden gesucht.

```
void GetDbBook  
(  
    String strText  
)
```

Parameter

strText
Suchkriterium

Rückgabewert

keiner

Bemerkungen

Die Funktion sucht in der Datenbank für die Rufnummernidentifikation nach Rufnummern. Hier sind Namen oder Namensfragmente gegeben und Datensätze mit Namen und Rufnummern werden zurückgeliefert. Ein "Sch" zum Beispiel sucht nach allen Namen wie Schmidt oder Schulz. Wird ein Leerstring übergeben, so erstellt die Datenbank ein komplettes Telefonbuch und schickt es an den Client zurück. Die zurückgesendeten Daten lösen im Client einen Event vom Typ [OnDbBookRecord](#) aus. Bevor diese Funktion genutzt werden kann ist es erforderlich dass die Datenbankschnittstelle des CTI-Servers eingerichtet wurde. Die Rufnummernerkennung auf der Seite "Datenbankschnittstelle" der Serverkonsole sollte aktiviert und eingerichtet sein.



Rufnummernerkennung / Datenbanktelefonbuch nutzen

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.4 Login

Die Anmeldung an die Datenbank erfolgt.

```
void Login  
(  
    String strPwd  
)
```

Parameter

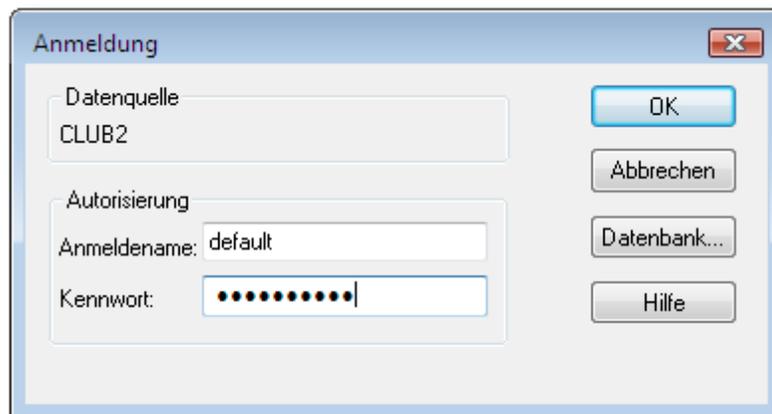
strPwd
Paßwort der Datenbank

Rückgabewert

keiner

Bemerkungen

Um die SQL-Schnittstelle nutzen zu können, meldet sich der Client mit dem Paßwort der Datenbank an. Das Paßwort wurde vom Administrator beim Einrichten der ODBC-Datenquelle vergeben.



bstrPwd kann auch ein leerer String sein, wenn kein Paßwort vergeben wurde. Als Antwort darauf erhält der Client ein Event vom Typ [OnSQLLogin](#).

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.5 Next

SQL-Abfrage wird fortgesetzt.

```
bool Next  
(  
    uint dwRequest,  
    uint dwCount  
)
```

Parameter

dwRequest
Abfrage-ID

dwCount
Maximale Anzahl von Datensätzen

Rückgabewert

Rückgabewert, True bei Erfolg

Bemerkungen

Eine mit [SQLBegin](#) gestartete Abfrage wird fortgesetzt. Dazu wird die Abfrage-ID übergeben, welche SQLBegin zurückgeliefert hat. Die Anzahl der maximal zu liefernden Datensätze wird begrenzt auf so viele Datensätze, wie in nächster Zeit sinnvoll weiterverarbeitet werden kann. Die Lebensdauer einer Abfrage verlängert sich durch jedes **SQLNext**. Ist der Client nicht an die Datenbank angemeldet, so schlägt die Funktion fehl und liefert False zurück.

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.6 OnDbBookRecord

Ein Datensatz aus der Rufnummernsuche wird übermittelt.

```
Event OnDbBookRecord
(
    object sender,
    String e.strName,
    String e.strNumber,
    String e.strExtra
)
```

Parameter

sender

Objekt [SQLTool](#)

e.strName

Name des Teilnehmers

e.strNumber

Seine Rufnummer

e.strExtra

Zusätzliche Informationen

Bemerkungen

Der Benutzer hat eine Anfrage an die Datenbank vom Typ [GetDbBook](#) gestartet. Die Datensätze kommen Record für Record. bstrExtra kann leer sein. Umfang und Format der ankommenden Daten werden in den Datenbankeinstellungen des CTI-Servers festgelegt.

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.7 OnSQLData

SQL-Abfragen werden beantwortet.

```
Event OnSQLData
(  
    object sender,  
    uint e.dwRequest,  
    int e.nCount,  
    String e.strData  
)
```

Parameter

sender

Objekt [SQLTool](#)

e.dwRequest

Nummer der Anfrage, welche mit [Begin](#) gestartet wurde.

e.nCount

Anzahl der zurückgelieferten Datensätze, ist 0 wenn keine Daten vorhanden waren und -1 wenn die Abfrage fehlschlug.

e.strData

Text mit den angeforderten Daten.

Bemerkungen

Dieser Event wird immer dann aufgerufen wenn Daten mit [SQLBegin](#) oder [Next](#) angefordert wurden. Die Daten können im XML-Format oder als Text kommen. Die Abfrage kann fehlschlagen wenn eine Abfrage bereits terminiert wurde oder bereits alle Daten gelesen wurden.

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.8 OnSQLEnd

Eine SQL-Abfrage wurde beendet.

```
Event OnSQLEnd
(  
    object sender,  
    uint e.dwRequest,  
    Bool e.bSuccess  
)
```

Parameter

sender

Objekt [SQLTool](#)

e.dwRequest

Nummer der Datenbankabfrage, welche bei [Begin](#) vergeben wurde.

e.bSuccess

Ist True wenn die Abfrage beendet werden konnte.

Bemerkungen

Der Event wird auf den Befehl [End](#) hin gesendet. Werden mit einem SQLBegin oder [Next](#) alle Datensätze gelesen, so kommt dieser Event selbständig. Wird eine Datenbankabfrage nicht terminiert, so terminiert der Server diese automatisch nach einigen Minuten und sendet diesen Event.

Siehe auch

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.4.9 OnSQLLogin

Die Anmeldung an die Datenbank wird bestätigt.

```
Event OnSQLLogin  
(  
    object sender,  
    Bool e.bSuccess  
)
```

Parameter

sender

Objekt [SQLTool](#)

e.bSuccess

Ist True, wenn die Anmeldung erfolgreich war.

Bemerkungen

Bevor ein Anwender die Datenbank benutzt muss er sich dort mit [Login](#) anmelden. Der Server schickt daraufhin diese Nachricht zurück. Wenn bei der Anmeldung ein falsches Paßwort übermittelt wurde oder die Datenbankverbindung nicht aktiv ist wird die Anmeldung abgewiesen. Die Datenbankverbindung wird auf der Serverkonsole auf der Seite "Datenbankschnittstelle" zugeschalten.

**Siehe auch**

[Übersicht CSLTool](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5 Telephony

Klasse Telephony

Die Klasse enthält alle Funktionen und Events welche mit den Gesprächen direkt zu tun haben.

Funktionen

AddCallToConference	Fügt ein Gespräch zu einer Konferenz hinzu.
AddNumberToConference	Tätigt einen Anruf und holt diesen in die Konferenz.
Alternate	Wechselt zwischen zwei Anrufen hin und her.
Answer	Nimmt ein Gespräch entgegen.
DialKey	Die Taste zum Anwählen von markiertem Text wird festgelegt.
DialSpecialKey	Die zusätzliche Steuertaste zum Anwählen von markiertem Text wird festgelegt.
DropKey	Die Taste zum Auflegen von Gesprächen wird festgelegt.
DropSpecialKey	Die zusätzliche Steuertaste zum Auflegen von Gesprächen wird festgelegt.
ForwardingDest	Das Rufumleitungsziel wird bearbeitet.
ForwardingState	Der Status der Rufumleitung wird bearbeitet.
Hangup	Ein Anruf wird aufgehängt.
Hold	Ein Gespräch wird gehalten.
LastDialedNumber	Die zuletzt angerufene Nummer.
LineReset	Die TAPI-Leitung wird zurückgesetzt.
MakeCall	Ein Anruf wird gestartet.
OnAcdCancelClient	Bei einem Client wurde der ACD Login Modus zurückgesetzt.
OnCallNumber	Eine Rufnummer wird übertragen.
OnCallReminder	Eine Anruferinnerung wurde dem Benutzer zugesendet.
OnCallState	Übermittelt den Zustand eigener Gespräche.
OnConfNotify	Die Konferenzzugehörigkeit hat sich geändert.
OnForwardingState	Die eigene Rufumleitung wird übertragen.
OnGroupMember	Teilt den Clients beim Start die Mitglieder ihrer Gruppe und deren Zustand mit.
OnIdentity	Die Rufnummernidentifikation kommt an.
OnObjectOfRedirect	Auf den Benutzer wird eine Rufumleitung geschaltet.
OnProtocolData	Gesprächsdatensätze werden übermittelt.
OnSendAddress	Die Adresse, unter welcher der Anruf einging wird gesendet.
OnSendUserState	Der Status eines Gruppenmitglieds hat sich geändert.

OnTransferPendInfo	Der interne Gesprächspartner hat einen wartenden Anrufer in der Leitung.
Pickup	Ein eingehendes, nicht entgegengenommenes Gespräch wird herangeholt.
PrivateState	Zeigt den Privatgesprächs-Modus an und ändert ihn.
ProxyMakeCall	Ein Anruf wird über ein Proxytelefon gestartet. Nützlich bei einigen DECT Telefonen.
ReDial	Die Wahlwiederholung wird gestartet.
Redirect	Ein Gespräch wird umgelenkt.
RemovePhonelcon	Das gelbe Telefonsymbol wird entfernt.
SendCallReminder	Eine Anruferinnerung wird gesendet.
SetAcdReady	ACD Status schalten.
SetForwarding	Die Rufumleitung wird zu- oder abgeschaltet.
Transfer	Gesprächsübergabe zweier Anrufe.
UnHold	Ein gehaltenes Gespräch wird zurückgeholt.
UserCallInfo	Die Art der Benachrichtigung bei eingehenden Anrufen wird bestimmt.

Siehe auch[Übersicht .NET](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.1 AddCallToConference

Fügt ein Gespräch zu einer Konferenz hinzu.

```
void AddCallToConference  
(  
    uint dwCallID  
)
```

Parameter

dwCallID

Anruf-ID des Gesprächs, welches zu einer Konferenz hinzugefügt werden soll. Wenn hier 0 steht dann versucht die CTI-Engine automatisch entweder eine Konferenz aufzubauen oder den wahrscheinlichsten Teilnehmer hinzuzufügen.

Rückgabewert

keiner

Bemerkungen

Mit dieser Funktion können Sie bereits bestehende Gespräche zu einer Konferenz hinzufügen. In der Regel werden Sie ein gehaltenes Gespräch nach einer Rückfrage mit in das aktive Gespräch holen. Nicht alle Anlagen unterstützen diese Funktion. Insbesondere Konferenzen mit mehr als drei Teilnehmern werden nur von wenigen Herstellern unterstützt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.2 AddNumberToConference

Tätigt einen Anruf und holt diesen in die Konferenz.

```
void AddNumberToConference  
(  
    String strNumber  
)
```

Parameter

strNumber

Rufnummer des Teilnehmers, welcher angerufen und in die Konferenz geholt werden soll.

Rückgabewert

keiner

Bemerkungen

Die Funktion ruft *strNumber* an. Nimmt dieser das Gespräch entgegen, so wird er in die Konferenz mit einbezogen bzw. das aktive Gespräch zu einer Dreierkonferenz ausgebaut. Nicht alle Anlagen unterstützen diese Funktion. Insbesondere Konferenzen mit mehr als drei Teilnehmern werden nur von wenigen Herstellern unterstützt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.3 Alternate

Wechselt zwischen zwei Anrufen hin und her.

```
void Alternate  
(  
    uint dwCall1,  
    uint dwCall2  
);
```

Parameter

dwCall1

Anruf-ID des Gesprächs, das gerade verbunden ist und ins Halten gestellt werden soll.

dwCall2

Anruf-ID des Gesprächs, welches gerade gehalten wird und verbunden werden soll.

Rückgabewert

keiner

Bemerkungen

Die Funktion wechselt zwischen zwei Anrufen hin und her. Ein Gespräch ist jeweils verbunden, das andere Gespräch wird gehalten. Die Parameter können auch vertauscht sein, die Funktion ist fehlertolerant. Sind die Parameter 0 dann Sucht die CTI-Engine automatisch nach den beiden zu tauschenden Anrufen. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.4 Answer

Nimmt ein Gespräch entgegen.

```
void Answer  
(  
    uint dwCallID  
)
```

Parameter

dwCallID

Anruf-ID des Gesprächs, das entgegengenommen werden soll. Wird 0 übergeben, so wird versucht den ersten Anruf in der eigenen Warteschlange entgegen zunehmen, der sich im State [OFFERING](#) befindet.

Rückgabewert

keiner

Bemerkungen

Nimmt einen eingehenden Anruf entgegen. Die Funktion hat nur dann Erfolg, wenn die Telefone ins Freisprechen schalten können, die meisten ISDN- und Systemtelefone können dies.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.5 DialKey

Die Taste zum Anwählen von markiertem Text wird festgelegt.

Eigenschaft DialKey

Typ

int32, lesen und schreiben

Bemerkungen

Ist [HotKey](#) aktiv geschaltet, so kann markierter Text mit einer F-Taste gewählt werden. Der Wert entspricht dem virtuellen Keycode der Taste und kann von 0x70 (F1) bis 0x7B (F12) liegen. Die F-Taste kann mit der Steuertaste und Umschalttaste kombiniert werden.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.6 DialSpecialKey

Die zusätzliche Steuertaste zum Anwählen von markiertem Text wird festgelegt.

Eigenschaft DialSpecialKey**Typ**

int32, lesen und schreiben

Bemerkungen

Die Steuertaste zum Anrufen von Nummern in markiertem Text kann noch mit Sondertasten kombiniert werden, die in dieser Eigenschaft festgelegt werden. Mögliche Werte sind:

0 = Keine zusätzliche Taste

1 = Umschalttaste

2 = Steuertaste

3 = Umschalt- und Steuertaste

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.7 DropKey

Die Taste zum Auflegen von Gesprächen wird festgelegt.

Eigenschaft DropKey

Typ

int32, lesen und schreiben

Bemerkungen

Ist [HotKey](#) aktiv geschaltet, so kann ein Gespräch mit einer F-Taste aufgelegt werden. Der Wert entspricht dem virtuellen Keycode der Taste und kann von 0x70 (F1) bis 0x7B (F12) liegen. Die F-Taste kann mit der Steuertaste und Umschalttaste kombiniert werden. Die Werte sollten sich von [DialKey](#) unterscheiden.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.8 DropSpecialKey

Die zusätzliche Steuertaste zum Auflegen von Gesprächen wird festgelegt.

Eigenschaft DropSpecialKey**Typ**

int32, lesen und schreiben

Bemerkungen

Die Steuertaste zum Auflegen von Gesprächen kann noch mit Sondertasten kombiniert werden, die in dieser Eigenschaft festgelegt werden. Mögliche Werte sind:

- 0 = Keine zusätzliche Taste
- 1 = Umschalttaste
- 2 = Steuertaste
- 3 = Umschalt- und Steuertaste

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.9 ForwardingDest

Das Rufumleitungsziel wird bearbeitet.

Eigenschaft ForwardingDest

Typ

String, lesen und schreiben

Bemerkungen

Das Rufumleitungsziel kann abgefragt werden. Wird die Eigenschaft geändert dann wird eine neue Rufumleitung mit dem geänderten Ziel aktiv.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.10 ForwardingState

Der Status der Rufumleitung wird bearbeitet.

Eigenschaft ForwardingState**Typ**

bool, lesen und schreiben

Bemerkungen

Schaltet die Rufumleitung zu oder ab ohne das Rufumleitungsziel zu verändern oder liefert den Status der Rufumleitung zurück.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.11 Hangup

Ein Anruf wird aufgehängt.

```
void Hangup  
(  
    uint dwCallID  
);
```

Parameter

dwCallID

Anruf-ID des Gesprächs, welches aufgehängt werden soll. Wird 0 übergeben, so wird das erste Gespräch in der Leitung aufgehängt.

Rückgabewert

keiner

Bemerkungen

Legt ein Gespräch auf. Ist in fast allen Treibern implementiert.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.12 Hold

Ein Gespräch wird gehalten.

```
void Hold  
(  
    uint dwCallID  
)
```

Parameter

dwCallID

Anruf-ID des Gesprächs, das ins Halten gestellt werden soll. Wenn diese 0 ist dann wird der wahrscheinlichste Anruf gehalten.

Rückgabewert

keiner

Bemerkungen

Stellt einen aktiven Anruf in den gehaltenen Zustand. Der Anrufer hört Wartemusik. Mit [UnHold](#) kann der Anruf zurückgeholt werden. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.13 LastDialedNumber

Die zuletzt angerufene Nummer.

Eigenschaft LastDialedNumber

Typ

String, nur lesen

Bemerkungen

Liefert die zuletzt gewählte Nummer.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.14 LineReset

Die TAPI-Leitung wird zurückgesetzt.

void LineReset()**Parameter**

Keine

Rückgabewert

keiner

Bemerkungen

Mitunter kann es vorkommen, dass der TAPI-Treiber nicht mehr reagiert. Diese Funktion setzt die Leitung zurück, in den meisten Fällen sollte dann das Problem behoben sein. Sie haben auch die Möglichkeit, im CTI-Server eine Autokorrektur zuzuschalten, welche versucht solche Probleme zu beheben. In der Serverkonsole finden Sie auf der Seite "TAPI-Sonderbehandlung" dazu die Möglichkeit.

**Siehe auch**

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.15 MakeCall

Ein Anruf wird gestartet.

```
void MakeCall  
(  
    String strNumber  
);
```

Parameter

strNumber
Rufnummer der angerufenen Stelle

Rückgabewert

keiner

Bemerkungen

Sollte bereits ein aktives Gespräch bestehen, wird versucht dieses ins Halten zu stellen und einen neuen Anruf zu tätigen. Sie können diese Funktion auch für Rückfragen einsetzen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.16 OnAcidCancelClient

Bei einem Client wurde der ACD Login Modus zurückgesetzt.

Event OnAcidCancelClient

```
(  
    object sender  
)
```

Parameter

sender

Objekt [Telephony](#)

Bemerkungen

Der Client wird auf dem Server aus der Liste der anwesenden Agents ausgetragen wenn er Gespräche nicht annimmt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.17 OnCallNumber

Eine Rufnummer wird übertragen.

```
Event OnCallNumber
(
    object sender,
    int e.nType,
    zunt e.dwCallID
    String e.strNumber
)
```

Parameter

sender

Objekt [Telephony](#)

e.nType

Art des Anrufs. Folgende Werte sind möglich:

- 1: Primäranruf eingehend
- 2: Anklopfender Anruf bei bestehendem Gespräch
- 3: Ausgehender Anruf

e.dwCallID

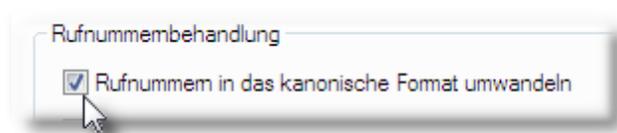
Anruf-ID

e.strNumber

Übertragene Rufnummer, maximal 31 Stellen.

Bemerkungen

Übermittelt Rufnummern, sofern diese mit übertragen wurden. Bei Anrufen aus dem Analognetz und anonymen Anrufen wird keine Rufnummer übertragen. Bei ausgehenden Rufen wird dieser Event ebenfalls ausgelöst. Das Format der Rufnummer kann auf dem Server als kanonisch festgelegt werden (Seite Anlageneinstellungen).



Normalerweise macht ein neuer Anruf auf sich aufmerksam, indem ein Event [OnCallState](#) mit dem Status [LINECALLSTATE_OFFERING](#) (eingehend) oder [LINECALLSTATE_DIALTONE](#) (ausgehend) auftritt. OnCallNumber kann jedoch auch vor allen anderen Events gesendet werden. Die Rufnummer kann sich während des Gesprächs ändern, wenn z. B. jemand dem Benutzer einen Anruf weitervermittelt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.18 OnCallReminder

Eine Anruferinnerung wurde dem Benutzer zugesendet.

```
Event OnCallReminder
(  
    object sender,  
    String e.bstrUser,  
    String e.strNumber,  
    String e.strMessage  
)
```

Parameter

sender

Objekt [Telephony](#)

e.strUser

Durchwahl des Benutzers, der die Anruferinnerung gesendet hat.

e.strNumber

Nummer, welche zurückgerufen werden soll.

e.strMessage

Zusätzliche Nachricht zum Problem. Hier könnte der Absender erklären, warum *bstrNumber* zurückgerufen werden soll.

Bemerkungen

Die Benutzer können sich gegenseitig Anruferinnerungen schicken. Somit muss man nicht mehr jedem mitteilen, wer dem zurückrufen soll, der Vorgang wird wesentlich kürzer gestaltet. Anruferinnerungen werden mit [TSendCallReminder](#) verschickt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.19 OnCallState

Übermittelt den Zustand eigener Gespräche.

```
Event OnCallState
(  
    object sender,  
    uint e.dwState,  
    uint e.dwCallID  
)
```

Parameter

sender

Objekt [Telephony](#)

e.dwState

[Gesprächszustand](#), in dem sich der Anruf befindet.

e.dwCallID

Anruf-ID, dessen Zustand sich ändert.

Bemerkungen

Der Zustand eines Gesprächs hat sich geändert. Diese Nachricht wird für die Gespräche des jeweiligen Benutzers übertragen, der das Gespräch führt. Sie sollten eine Liste mit aktiven Anrufen im Programm mitführen. Prüfen Sie bei jedem Aufruf von **OnCallState** ob die Anruf-ID in der Liste enthalten ist und fügen Sie diese bei Bedarf ein. Sollte ein Status [LINECALLSTATE_IDLE](#) gesendet werden, so entfernen Sie den Anruf aus der Liste. In der Liste werden sich meist nur ein oder zwei Anrufe befinden.

Zustände der Gruppenmitglieder werden mit dem Event [OnSendUserState](#) gesendet.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.20 OnConfNotify

Die Konferenzzugehörigkeit hat sich geändert.

Event OnConfNotify

```
(  
    object sender,  
    bool e.bConf  
)
```

Parameter

sender

Objekt [Telephony](#)

e.bConf

True, wenn der Benutzer in eine Konferenz eintritt, False, wenn er sie verlässt.

Bemerkungen

Beim Eintreten in eine und beim Verlassen von Konferenzen wird diese Nachricht gesendet.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.21 OnForwardingState

Die eigene Rufumleitung wird übertragen.

```
Event OnForwardingState
(  
    object sender,  
    bool e.bSet,  
    String e.strNumber  
)
```

Parameter

sender

Objekt [Telephony](#)

e.bSet

Wenn True, dann ist eine Rufumleitung geschaltet.

e.strNumber

Nummer, zu der umgeleitet wird.

Bemerkungen

Die Einstellungen für die Rufumleitungen werden zentral auf dem CTI-Server gespeichert. Deshalb wird diese Nachricht nach der Anmeldung gesendet. Bei Änderungen an der Rufumleitung wird **OnForwardingState** ebenfalls aufgerufen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.22 OnGroupMember

Teilt den Clients beim Start die Mitglieder ihrer Gruppe und deren Zustand mit.

```
Event OnGroupMember
(  
    object sender,  
    String e.strNumber,  
    String e.strName,  
    bool e.bOnline,  
    bool e.bActiveCall,  
    bool e.bRinging,  
    bool e.bRedirection  
)
```

Parameter

sender

Objekt [Telephony](#)

e.strNumber

Durchwahl des Gruppenmitglieds, 2 bis 4 Stellen.

e.strName

Name des Gruppenmitglieds. Hier wird der Anmeldename verwendet. Dieser ist erst verfügbar wenn sich dieser Kollege mindestens einmal eingeloggt hat, da sich diese Daten automatisch einpflegen.

e.bOnline

Ist True, wenn der Teilnehmer gerade eingeloggt ist.

e.bActiveCall

Ist True, wenn der Teilnehmer gerade aktive Gespräche hat.

e.bRinging

Ist True, wenn unbeantwortete Anrufe beim Teilnehmer in der Warteschlange stehen. Diese Anrufe können bei vielen Anlagen herangeholt werden.

e.bRedirection

Ist True, wenn der Teilnehmer eine Rufumleitung geschaltet hat.

Bemerkungen

Bei jedem Anmelden werden die Daten der Gruppenmitglieder geschickt. Pro Gruppenmitglied wird eine Nachricht gesendet. Wird das Event mit einem leeren String in *strName* aufgerufen, so ist die Übertragung beendet. Mit diesen Events sind gleich zu Programmstart die Gesprächszustände der Gruppenmitglieder bekannt. Diese werden zur Laufzeit mit [OnSendUserState](#) aktualisiert.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.23 OnIdentity

Die Rufnummernidentifikation kommt an.

```
Event OnIdentity  
(  
    object sender,  
    uint e.dwCallID,  
    String e.strName  
)
```

Parameter

sender

Objekt [Telephony](#)

e.dwCallID

Anruf-ID des Gesprächs

e.strName

Name des Anrufers, bis zu 255 Zeichen

Bemerkungen

Auf dem Server wurde eine Rufnummernidentifikation durchgeführt. Diese kann von der ODBC-Schnittstelle oder der Server Programmierschnittstelle kommen. Neben dem Namen können in *strName* noch weitere Daten übertragen werden.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.24 OnObjectOfRedirect

Auf den Benutzer wird eine Rufumleitung geschaltet.

```
Event OnObjectOfRedirect  
(  
    object sender,  
    bool e.bSet,  
    String e.strNumber  
)
```

Parameter

sender

Objekt [Telephony](#)

e.bSet

Auf den Teilnehmer wurde eine Rufumleitung geschaltet, wenn True. Wenn False wurde die Rufumleitung wieder abgeschaltet.

e.strNumber

Durchwahl des Teilnehmers, der zu diesem Anschluß umleitet.

Bemerkungen

Wenn ein anderer Teilnehmer eine Rufumleitung hierher schaltet wird dieser Event aufgerufen. Ebenso wenn dieser die Rufumleitung wieder herausnimmt. Sollten Rufumleitungen auf diesen Anschluß geschaltet sein so wird die Meldung auch nach Anmeldung übertragen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.25 OnProtocolData

Gesprächsdatensätze werden übermittelt.

```
Event OnProtocolData
(
    object sender,
    String e.strOther,
    String e.strOwn,
    uint e.dwStart,
    uint e.dwEnd,
    bool e.bSuccess,
    bool e.blInbound
)
```

Parameter

sender

Objekt [Telephony](#)

e.strOther

Rufnummer des Gesprächspartners

e.strOwn

Eigene Durchwahl

e.dwStart

Gesprächsbeginn, Format time_t

e.dwEnd

Gesprächsende, Format time_t

e.bSuccess

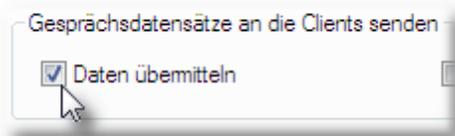
Ist True, wenn das Gespräch erfolgreich war, beide Partner waren miteinander verbunden.

e.blInbound

Ist True, wenn das Gespräch eingehend war, bei ausgehenden Gesprächen ist *blInbound* False.

Bemerkungen

Nach Ende jedes Gesprächs wird ein solcher Gesprächsdatensatz übertragen. Ist der Benutzer abgemeldet, so werden die Daten auf dem CTI-Server gepuffert und bei der nächsten Anmeldung gesendet. Die Daten werden nur gesendet, wenn auf dem Server diese Option eingeschaltet ist. Auf der Seite "Datenbankschnittstelle" der Serverkonsole kann die Übermittlung zugeschaltet werden.

**Siehe auch**

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.26 OnSendAddress

Die Adresse, unter welcher der Anruf einging wird gesendet.

```
Event OnSendAddress  
(  
    object sender,  
    bool e.blsAddress,  
    uint e.dwCallID,  
    String e.strAddress  
)
```

Parameter

sender

Objekt [Telephony](#)

e.blsAddress

Wenn True dann wird die Adresse des Telefons in *bstrAddress* übermittelt, bei False ist es die Rufnummer.

e.dwCallID

Anruf-ID des Gesprächs

e.strAddress

Adresse welche angerufen wurde oder Rufnummer.

Bemerkungen

Diese Nachricht wird bei eingehenden Anrufen gesendet. Ein ISDN-Telefon kann auf mehrere MSNs reagieren. Nehmen wir z. B. einen Heimarbeitsplatz. Hier wird es eine MSN für Privatgespräche und eine MSN für dienstliche Anrufe geben. Am Monatsende soll aber ausgewertet werden, wieviel Zeit dienstlich telefoniert wurde. Über die angerufene MSN in *bstrAddress* können die Anrufe zugeordnet werden.

Eine andere Möglichkeit trifft man bei Telefonanlagen. Eine Leitung kann mehrere interne Durchwahlen haben. Die 210 kann für normale und interne Gespräche verwendet werden, die 310 ist mit einer Hotline verbunden. In *strAddress* würde entweder die 210 oder die 310 mitgeliefert. TAPIMaster® versucht, eine der beiden Angaben zu ermitteln und zu senden. Adressen werden jedoch von nur von wenigen Treibern unterstützt. Was gesendet wird, ist von der Anlage bzw. dem Telefon und den Treibern abhängig. Es ist auch möglich, dass diese Nachricht nicht gesendet wird oder mehrmals während eines Anrufs kommt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.27 OnSendUserState

Der Status eines Gruppenmitglieds hat sich geändert.

```
Event OnSendUserState
(  
    object sender,  
    String e.strNumber,  
    bool e.bLogin,  
    bool e.bActiveCall,  
    bool e.bRinging,  
    bool e.bRedirection  
)
```

Parameter

sender

Objekt [Telephony](#)

e.strNumber

Durchwahl des Gruppenmitglieds

e.bLogin

Ist True wenn das Gruppenmitglied angemeldet ist.

e.bActiveCall

Ist True, wenn das Gruppenmitglied gerade ein Gespräch führt.

e.bRinging

Ist True wenn das Gruppenmitglied angerufen wird und diesen Anruf noch nicht angenommen hat.

e.bRedirection

Ist True, wenn das Gruppenmitglied Anrufe umleitet.

Bemerkungen

Die Nachricht wird übertragen, wenn sich der Zustand eines Gruppenmitglieds geändert hat. Der Benutzer hat somit die Möglichkeit zu reagieren wenn ein Gruppenmitglied ein Gespräch nicht annimmt oder ein anklopfendes Gespräch erhält. Seine Erreichbarkeit kann ebenfalls immer eingesehen werden.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.28 OnTransferPendInfo

Der interne Gesprächspartner hat einen wartenden Anrufer in der Leitung.

```
Event OnTransferPendInfo  
(  
    object sender,  
    String e.strMember,  
    String e.strHeldParty,
```

Parameter

sender

Objekt [Telephony](#)

e.strMember

Nebenstelle des Gesprächspartners

e.bLogin

Nummer des .wartenden Anrufers

Bemerkungen

In der Datei FCTIINI.TXT des Servers muss die Option

TransferPendInfo=1

gesetzt sein.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.29 Pickup

Ein eingehendes, nicht entgegengenommenes Gespräch wird herangeholt.

```
void Pickup  
(  
    String strNumber  
)
```

Parameter

strNumber

Durchwahl des Apparats, von dem ein Gespräch herangeholt werden soll.

Rückgabewert

keiner

Bemerkungen

Mit Hilfe dieser Funktion lassen sich Gespräche von anderen Apparaten heran holen. TAPIMaster® arbeitet hier teilweise mit Ersatzfunktionen gegenüber der TAPI, so dass auch ein anlagenübergreifendes Heranholen von Gesprächen möglich ist. Sollte die Funktion trotzdem fehlschlagen, kann TAPIMaster® direkt Hardware-Steuerbefehle absetzen um das Gespräch heranzuholen. Je nach Anlage können auch anklopfende Gespräche gepickt werden.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.30 PrivateState

Zeigt den Privatgesprächs-Modus an und ändert ihn.

Eigenschaft PrivateState

Typ

bool, lesen und schreiben

Bemerkungen

Für den Privatmodus können andere Amtskennungen definiert worden sein. Diese werden dann verwendet.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.31 ProxyMakeCall

Ein Anruf wird über ein Proxytelefon gestartet. Nützlich bei einigen DECT Telefonen.

```
void ProxyMakeCall  
(  
    String strNumber,  
    String strProxy,  
);
```

Parameter

strNumber

Rufnummer der angerufenen Stelle

strProxy

Durchwahl des vermittelnden Telefons, muss ein Systemtelefon sein.

Rückgabewert

keiner

Bemerkungen

Ein Teilnehmer kann nicht selbst über TAPI telefonieren. Ein Vermittlertelefon ruft den Akteur an. Der Akteur hebt ab. Daraufhin startet das Vermittlertelefon eine Rückfrage zum eigentlichen Anrufziel. Das Gespräch wird noch in der Anwahl übergeben.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.32 ReDial

Die Wahlwiederholung wird gestartet.

void ReDial()**Parameter**

Keine

Rückgabewert

keiner

Bemerkungen

Tätigt eine Wahlwiederholung der zuletzt angerufenen Nummer.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.33 Redirect

Ein Gespräch wird umgelenkt.

```
void CallRedirect  
(  
    uint dwCallID  
    String strNumber  
)
```

Parameter

dwCallID

Anruf-ID des Gesprächs, das umgeleitet werden soll.

strNumber

Zielnummer, zu der ein Anruf umgeleitet werden soll.

Rückgabewert

keiner

Bemerkungen

Die Funktion leitet Anrufe zu anderen Teilnehmern weiter. Es ist gleichgültig, ob der Anruf bereits verbunden ist oder noch nicht entgegengenommen wurde. Bei guten Anlagen können auch anklopfende Gespräche weitergeleitet werden. Die Funktion wird nicht von allen Telefonanlagen unterstützt. Wenn ICallID Null ist dann wird der wahrscheinlichste Anruf im State OFFERING / ACCEPTED weitergeleitet. Ist kein Anruf in dem State dann ist das wahrscheinlichste verbundene Gespräch Gegenstand der Weiterleitung.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.34 RemovePhonelcon

Das gelbe Telefonsymbol wird entfernt.

void RemovePhonelcon()**Parameter**

Keine

Rückgabewert

keiner

Bemerkungen

Bei eingehenden Anrufen kann ein kleines Telefonsymbol eingeblendet sein, wenn die Option [Phonelcon](#) aktiviert wurde. Mit dieser Funktion kann es wieder entfernt werden wenn das Telefonat bearbeitet ist.

**Siehe auch**

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.35 SendCallReminder

Eine Anruferinnerung wird gesendet.

```
void SendCallReminder  
(  
    String strOther  
    String strNumber  
    String strInfo  
)
```

Parameter

strOther

Durchwahl des Teilnehmers, welcher informiert werden soll.

strNumber

Rufnummer des Anrufers, der zurückgerufen werden soll.

strInfo

Kurze Nachricht (bis 255 Zeichen), die das Problem des Anrufers beschreiben.

Rückgabewert

keiner

Bemerkungen

Ein Anruf kommt herein. Es stellt sich heraus dass der Anrufer einen Kollegen sprechen wollte der aber gerade nicht da ist. Mit dieser Funktion kann der Kollege schnell und einfach über den Anruf informiert werden. Es wird ein Event vom Typ [OnCallReminder](#) im Client des Kollegen ausgelöst. Ist dieser nicht angemeldet, erhält er den Event bei seiner Anmeldung. Somit gehen weniger Informationen verloren, die Daten können auf dem Zielclient gleich für eine Anrufvorlage genutzt werden, zeitraubendes Abschicken von Mails entfällt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.36 SetAcidReady

ACD Status schalten.

```
void SetAcidReady  
(  
    bool bReady  
)
```

Parameter

bReady
Zu- oder Abschalten

Rückgabewert

keiner

Bemerkungen

Der Client teilt dem Server mit ob er als Agent für das ACD Light Modul zur Verfügung steht.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.37 SetForwarding

Die Rufumleitung wird zu- oder abgeschaltet.

```
void SetForwarding  
(  
    bool bSet,  
    String strNumber  
)
```

Parameter

bSet

Zu- oder Abschalten der Umleitung

strNumber

Nummer, zu der umgeleitet werden soll.

Rückgabewert

keiner

Bemerkungen

Schaltet eine Rufumleitung oder zu oder ab. Als Antwort darauf erhält der Teilnehmer einen Event vom Typ [OnForwardingState](#). Die Funktion kann fehlschlagen, wenn die Zielnummer bereits umgeleitet hat, die Rechte zur Umleitung nicht vorliegen oder die TAPI-Schnittstelle der Anlage die Funktion nicht unterstützt. Im letzten Fall gibt es die Möglichkeit, die Rufumleitung über Telefoncodes zu steuern, diese werden in der Serverkonsole auf der Seite "TAPI-Sonderbehandlungen" eingestellt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.38 Transfer

Gesprächsübergabe zweier Anrufe.

```
void Transfer  
(  
    uint dwCall1,  
    uint dwCall2  
)
```

Parameter

dwCall1
Anruf-ID des ersten Anrufs

dwCall2
Anruf-ID des zweiten Anrufs

Rückgabewert

keiner

Bemerkungen

Zwei Anrufe sollen miteinander verbunden werden. Meist ist ein Gespräch davon aktiv und das andere gehalten (hört Wartemusik). Die Reihenfolge in der die Anruf-IDs übergeben werden spielt keine Rolle. Die Funktion wird nicht von allen Telefonanlagen unterstützt. Ist einer oder beide Parameter 0 dann sucht die CTI-Engine automatisch nach den beiden Anrufen und verbindet die mit der größten Wahrscheinlichkeit.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.39 UnHold

Ein gehaltenes Gespräch wird zurückgeholt.

```
void CallUnHold  
(  
    uint dwCallID  
)
```

Parameter

dwCallID

Anruf-ID des Gesprächs, welches wieder verbunden werden soll. Ist der Parameter 0 dann wird das letzte gehaltene Gespräch zurückgeholt.

Rückgabewert

keiner

Bemerkungen

Ein mit [TCallHold](#) gehaltener Anruf kann zurückgeholt werden, so dass das Gespräch wieder aktiv ist. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.1.5.40 UserCallInfo

Die Art der Benachrichtigung bei eingehenden Anrufen wird bestimmt.

Eigenschaft OUserCallInfo**Typ**

int32, lesen und schreiben

Bemerkungen

Legt fest wie der an der Schnittstelle angedockte Client über einen eingehenden Anruf informiert werden soll. Mögliche Werte sind:

- 0 Keine Information
- 1 Der Telefonclient erhält ein Event, worauf er sich nach vorne bringen soll
- 2 Kundenspezifische Anzeige

Die Kundenspezifische Anzeige wird im Standardclient verwendet.

Siehe auch

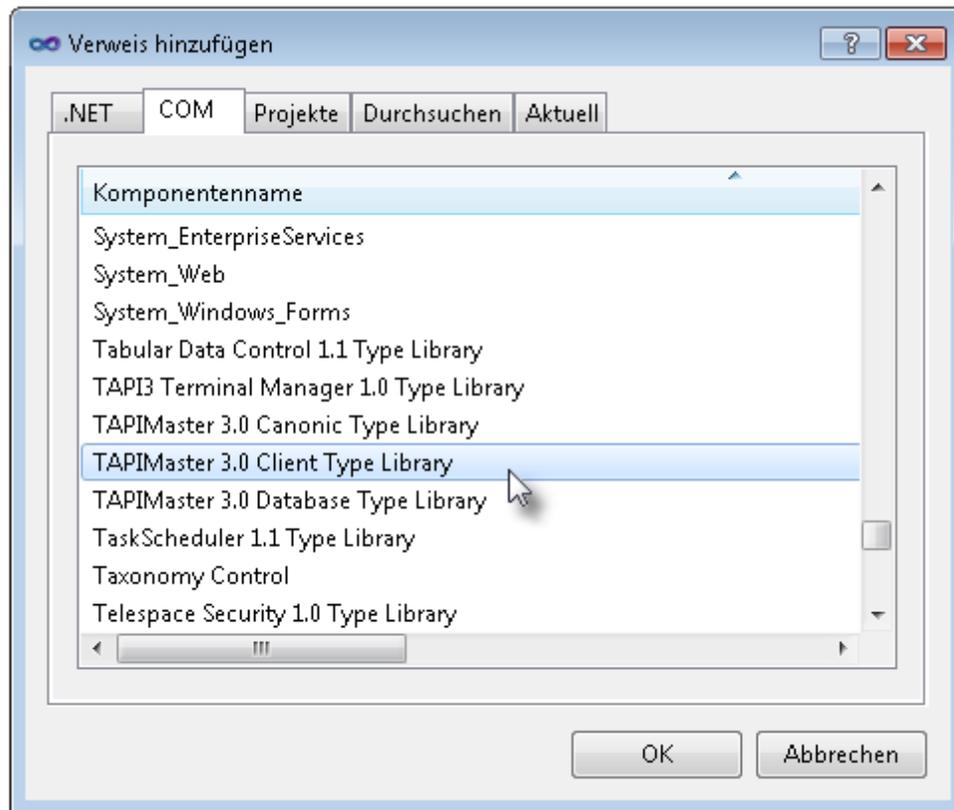
[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

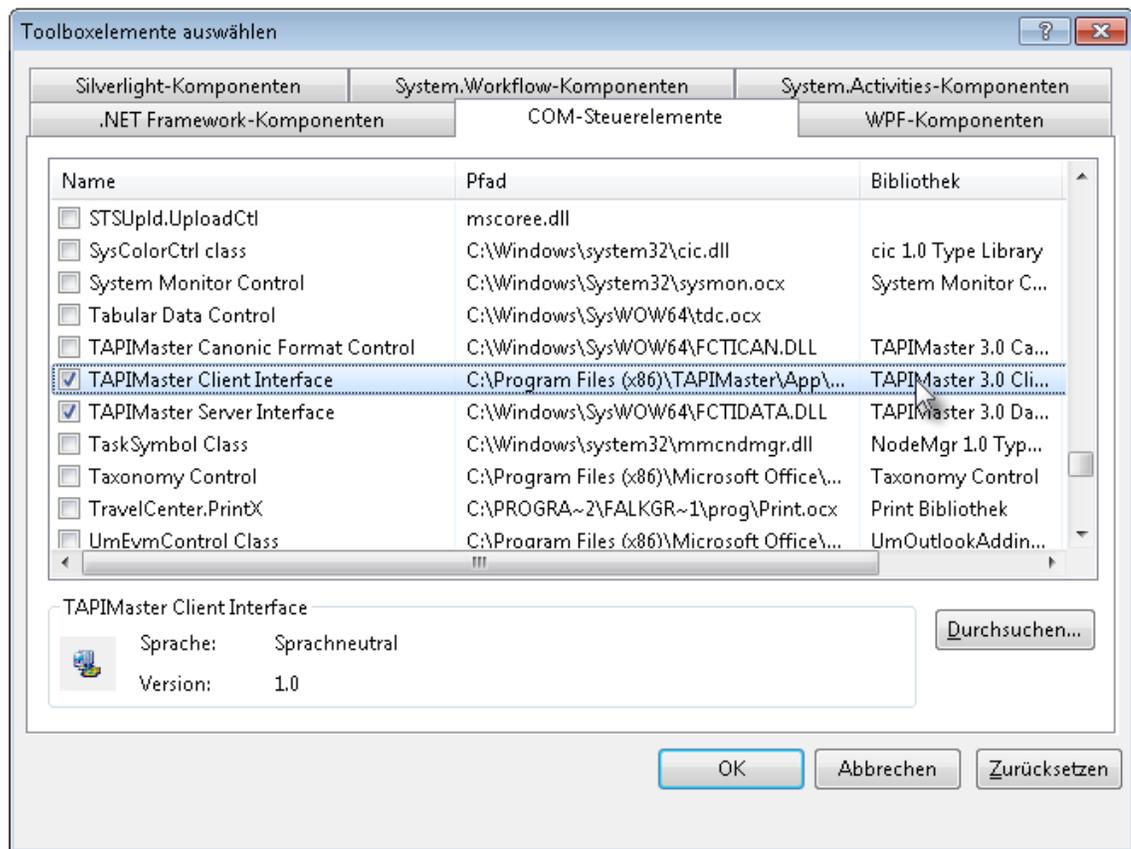
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2 ActiveX

Die Schnittstelle für ActiveX befindet sich in der Datei FCTICLNT.DLL. Führen Sie auf dem Zielrechner ein [Clientsetup](#) aus, damit dort alle notwendigen Dateien vorhanden bzw. registriert sind. Hier die Vorgehensweise für die Einbindung in VB / C#: Projekt | Verweis hinzufügen...



Jetzt kann das Objekt in die Toolbox übernommen werden. Anschließend wird es auf dem Formular platziert.



Die Funktionen sind nach Kategorien geordnet. Die wichtigsten Funktionen können Sie anhand der Beispiele für [VB](#) und [Delphi](#) im Einsatz sehen.

[Netzwerk](#)
[Optionen](#)
[Telefoniefunktionen](#)
[SQL](#)
[Zusatzfunktionen](#)
[Events](#)

Siehe auch

[Befehlsreferenz Client](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1 Netzwerk

Verwendung der Netzwerkfunktionen

Die Netzwerkfunktionen dienen der Steuerung der Verbindung zum CTI-Server.

Präfix

Die Netzwerkfunktionen haben den Präfix "N" vorangestellt.

Serveridentifikation

NServerName	Name oder IP-Adresse des CTI Servers
NPortNumber	Portnummer der Netzwerkverbindung
NNetworkConfigDialog	Zeigt einen Dialog mit den wichtigsten Netzwerk- und Verbindungsparametern an.

Steuerung der Netzwerkverbindung

NClientActive	Netzwerkverbindung aufbauen oder beenden.
NIsNetworkVersion	Ist die verwendete DLL die Netzwerkversion?
NLogin	Mit Benutzernamen am CTI Server anmelden.
NLogoff	Vom CTI Server abmelden.
NPrepareRestart	Neustart des Clients vorbereiten.
NStartClient	Initialisierung der Schnittstelle

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.1 NClientActive

Schaltet Netzwerkverbindung zu oder ab.

Eigenschaft NClientActive**Typ**

Bool, lesen und schreiben

Bemerkungen

Setzen Sie diese Eigenschaft auf False wenn Sie eine bestehende Netzwerkverbindung zum CTI-Server beenden möchten. Ein Wert von True stellt die Verbindung wieder her.

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [NStartClient](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.2 NlsNetworkVersion

Info über den Typ der FCTICLNT.DLL.

Eigenschaft NlsNetworkVersion**Typ**

Bool, nur lesen

Bemerkungen

Zeigt an ob es sich um die Netzwerkversion der FCTICLNT.DLL handelt. Das wird immer der Fall sein. Die Eigenschaft ist aus Kompatibilitätsgründen vorhanden.

Siehe auch

[Übersicht Netzwerkfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.3 NLogin

Meldet Client am CTI-Server an.

Funktion NLogin()**Parameter**

Keine

Bemerkungen

Die Funktion wird automatisch ausgeführt, wenn [OAutoLogin](#) aktiv ist und die Netzwerkverbindung steht. **NLogin** schaltet im CTI-Server die Funktionalität für den Client frei. Die anderen Gruppenmitglieder sehen den Client dann als angemeldet. Wenn Sie die Schnittstelle in ein Programm einbinden was den ganzen Tag über läuft kann es sinnvoll sein die CTI-Funktionalität zeitweilig abzuschalten, z. B. während der Mittagspause. Die anderen Gruppenmitglieder sehen dann dass der Benutzer nicht ansprechbar ist. [NLogoff](#) schaltet die CTI ab und **NLogin** wieder an.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [OAutoLogin](#) | [NLogoff](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.4 NLogoff

Meldet Client vom CTI-Server ab.

Funktion NLogoff()**Parameter**

Keine

Bemerkungen

Schaltet die CTI-Funktionalität im CTI-Server ab. Die Gruppenmitglieder sehen den Client dann als abgemeldet. Alternativ dazu kann auch die Netzwerkverbindung komplett mit gekappt werden wenn [NClientActive](#) auf False gesetzt wird.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [NLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.5 NNetworkConfigDialog

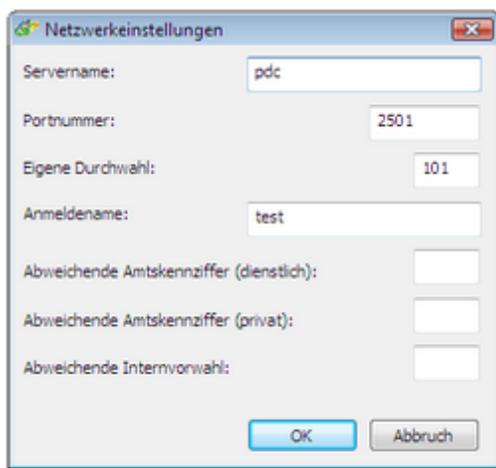
Zeigt Netzwerkdialog an.

Funktion NNetworkConfigDialog()**Parameter**

Keine

Bemerkungen

Ein Dialog wird geöffnet in welchem die Verbindungsdaten eingestellt werden können. Alternativ dazu ist die Verwendung von [NServerName](#), [NPortNumber](#) und [OOwnerNumber](#) möglich. Fehlen die ersten drei Angaben und ist die Eigenschaft [OCheckConnection](#) auf True gesetzt, so erscheint der Dialog automatisch beim Programmstart. Für die Anzeige des Dialogs wird die Verbindung zum CTI-Server unterbrochen. Mit OK wird eine neue Verbindung hergestellt. Benutzen Sie diesen Dialog wenn Sie nach einer Möglichkeit suchen die Verbindungsparameter zu ändern und selbst keine Eingabemöglichkeit programmieren wollen.

**Beispiel**

[Mini CTI Client \(Delphi\)](#)

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [NServerName](#) | [NPortNumber](#) | [OOwnerNumber](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.6 NPortNumber

Ändert Portnummer der Netzwerkverbindung.

Eigenschaft NPortNumber**Typ**

Long, lesen und schreiben

Bemerkungen

Die Portnummer der Verbindung. Wird der Wert geändert, so gibt es keinen Neuaufbau der Verbindung mit den geänderten Parametern. Um eine neue Netzwerkverbindung zu starten schalten Sie die Option [NClientActive](#) aus und wieder an.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.7 NPrepareRestart

Bereitet Client auf Neustart vor.

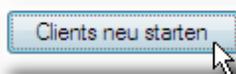
**Funktion NPrepareRestart(
Long IMinutes
)****Parameter**

IMinutes

Zeit in Minuten, die bis zum Neustart vergehen soll. Bei *IMinutes* = 0 erfolgt der Neustart sofort, bei negativen Werten wird der Client nicht neu gestartet.

Bemerkungen

Der CTI-Server kann die Standardclients herunterfahren und wieder neu starten. Das kann sinnvoll sein wenn die Clients aus einem Netzwerk heraus gestartet wurden und nun die Dateien ausgetauscht werden sollen. Über die Serverkonsole auf der Seite "Benutzerverwaltung" kann der Neustart angestoßen werden.



Sie können diese Funktionen ebenfalls unterstützen. Wenn ein Event [OnClientClose](#) eintritt, so bekommen Sie in dessen IParam die Zeit mitgeteilt, in welcher der Client neu starten soll. Rufen Sie

jetzt [NPrepareRestart](#) mit dieser Zeit als Parameter auf und beenden Sie die Anwendung. Sie wird nach der verstrichenen Zeit neu gestartet.

Siehe auch

[Übersicht Netzwerkfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.8 NServerName

Ändert den Servernamen der Netzwerkverbindung.

Eigenschaft NServerName**Typ**

String, lesen und schreiben

Bemerkungen

Der Name des CTI-Servers. Das kann auch die IP-Adresse oder "localhost" sein. Wird der Wert geändert, so gibt es keinen Neuaufbau der Verbindung mit den geänderten Parametern. Um eine neue Netzwerkverbindung zu starten schalten Sie die Option [NClientActive](#) aus und wieder an.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [NNetworkConfigDialog](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.1.9 NStartClient

Initialisiert der CTI-Schnittstelle.

Funktion NStartClient()**Parameter**

Keine

Bemerkungen

Mit dieser Funktion wird die [Schnittstelle](#) initialisiert. Ab nun können Events empfangen werden.

Beispiel

[Mini CTI Client \(Delphi\)](#)

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [NClientActive](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2 Optionen

Verwendung der Optionen

Die Optionen bestimmen das Verhalten der [CTI Schnittstelle](#). Sie werden geschaltet bevor die Schnittstelle mit [NStartClient](#) initialisiert wird.

Die Programmoptionen werden von der Schnittstelle intern gespeichert, es ist nicht erforderlich dass Sie diese Werte selbst sichern. Die Startwerte holt sich die Schnittstelle bei der [Installation](#) aus der Datei FCTIINST.INI, falls diese nicht vorhanden ist werden Voreinstellungen verwendet. Sie benötigen die Optionen normalerweise nicht, es sei denn Ihr Programm ermöglicht es diese zu ändern. Nutzen Sie die Schnittstelle für verschiedene CTI-Programme auf dem gleichen Rechner, so sollte jedes Programm seine Standardeinstellungen explizit setzen. Die meisten Optionen außer der internen Durchwahl können auch mit dem [Setup-Konfigurator](#) festgelegt werden. Er schreibt die FCTIINST.INI.

Präfix

Die Optionsfunktionen haben den Präfix "O" vorangestellt.

Schalter

OOptions	Vereinigt die übrigen Schalteroptionen als Bitflags in einem Wert.
OAssistedTelephony	Client stellt assisted Telephony bereit (reagiert auf den Aufruf von <code>tapiRequestMakeCall</code>).
OAutoAnswer	Eingehende Gespräche werden automatisch entgegengenommen.
OAutoConnect	Beim Programmstart wird selbständig die Netzwerkverbindung hergestellt.
OAutoLogin	Der Client meldet sich beim Start selbständig am CTI Server an.
OCheckConnection	Bringt einen Anmeldedialog nach vorn, wenn die Verbindungsparameter nicht komplett sind.
OHotKey	Verwendung von Schnellwahltasten
OPhoneIcon	Anrufsymbol wird angezeigt.
OShowIcon	Symbol für Netzwerkverbindung anzeigen.
OShowMenu	Benutzerdefiniertes Menü anzeigen.

Parameter

ODialKey	F-Taste für Anwahl
ODialSpecialKey	Zusatztaste für Anwahl
ODropKey	F-Taste zum Auflegen
<code>ODropKeyExtraState</code>	Veraltet. Benutzen Sie ODropSpecialKey .
<code>ODropKeyNumber</code>	Veraltet. Benutzen Sie ODropKey .
ODropSpecialKey	Zusatztaste zum Auflegen
<code>OHotKeyExtraState</code>	Veraltet. Benutzen Sie ODialSpecialKey .
<code>OHotKeyNumber</code>	Veraltet. Benutzen Sie ODialKey .

OIniFileName	Datei in der die CTI Schnittstelle ihre Einstellungen speichert
OOwnerName	Der Anmeldename wird festgelegt.
OOwnerNumber	Eigene Durchwahl
OUserCallInfo	Anzeige des Clients bei eingehenden Anrufen

Funktionen

OConnectionIconOptions	Benutzerdefiniertes Verbindungssymbol, Tooltip und Menü dazu
OShellIconOptions	Veraltet. Benutzen Sie OConnectionIconOptions .

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.1 OAssistedTelephony

Unterstützung von Assisted Telephony zuschalten.

Eigenschaft OAssistedTelephony

Typ

Bool, lesen und schreiben

Bemerkungen

Nimmt Befehle für Assisted Telephony Funktionen entgegen. Wenn aus einem Fremdprogramm die Windows TAPI Funktion `tapiRequestMakeCall` aufgerufen wird, so startet die Schnittstelle einen Anruf.

Voreinstellung

True

Siehe auch

[Übersicht Optionen](#) | [OOptions](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.2 OAutoAnswer

Gespräche werden automatisch angenommen.

Eigenschaft OAutoAnswer

Typ

Bool, lesen und schreiben

Bemerkungen

Eingehende Gespräche werden automatisch angenommen. Diese Option macht Sinn in Callcentern. Das Telefon / Headset muss ins Freisprechen schalten können und der TAPI Treiber dies unterstützen.

Voreinstellung

False

Siehe auch

[Übersicht Optionen](#) | [OOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.3 OAutoConnect

Die Netzwerkverbindung wird selbständig gehalten.

Eigenschaft OAutoConnect**Typ**

Bool, lesen und schreiben

Bemerkungen

Beim Programmstart stellt die CTI-Schnittstelle selbständig die Verbindung zum CTI-Server her. Wird diese Verbindung unterbrochen, so versucht der Client diese wieder aufzubauen. [NClientActive](#) muss nicht extra auf True gesetzt werden.

Voreinstellung

True

Beispiel

[Mini CTI Client \(Delphi\)](#)

Siehe auch

[Übersicht Optionen](#) | [OOptions](#) | [NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.4 OAutoLogin

Selbständig an CTI-Server anmelden.

Eigenschaft OAutoLogin**Typ**

Bool, lesen und schreiben

Bemerkungen

Die Schnittstelle meldet sich nach Aufbau der Netzwerkverbindung selbständig am Server an, [NLogin](#) muss nicht separat aufgerufen werden.

Beispiel

[Mini CTI Client \(Delphi\)](#)

Voreinstellung

True

Siehe auch

[Übersicht Optionen](#) | [OOptions](#) | [NLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.5 OCheckConnection

Die Verbindungsdaten werden beim Start geprüft.

Eigenschaft OCheckConnection**Typ**

Bool, lesen und schreiben

Bemerkungen

Beim Programmstart prüft die Schnittstelle die Verbindungsparameter auf Vollständigkeit. Das wären Servername, Portnummer und interne Durchwahl. Fehlt ein Parameter, so bringt die Schnittstelle den [Anmeldedialog](#) nach vorne.

Voreinstellung

True

Beispiel

[Mini CTI Client \(Delphi\)](#)

Siehe auch

[Übersicht Optionen](#) | [OOptions](#) | [NNetworkConfigDialog](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.6 OConnectionIconOptions

Das Aussehen des Verbindungsicons wird geändert.

Funktion OConnectionIconOptions(

```
    ULong hlconOn,  
    ULong hlconOff,  
    ULong hMenu,  
    String bstrText
```

```
);
```

Parameter

hlconOn

Handle des Icons, welches bei bestehender Netzwerkverbindung zum CTI Server angezeigt wird. Bei 0 wird das Standardicon angezeigt.

hlconOff

Handle des Icons, welches bei unterbrochener Netzwerkverbindung zum CTI Server angezeigt wird. Bei 0 wird das Standardicon angezeigt.

hMenu

Handle auf ein Menü, welches über dem Verbindungsicon angezeigt wird, wenn dieser mit der rechten Maustaste darauf klickt. Kann 0 sein, dann wird kein Menü verwendet.

bstrText

Tooltip, welcher über dem Verbindungsicon angezeigt wird. Maximal 63 Zeichen. Kann Null sein, dann wird ein Standardtext angezeigt. Soll kein Tooltip angezeigt werden, so übergeben Sie einen Leerstring.

Bemerkungen

Aussehen, Menü und Tooltips des Verbindungsicons lassen sich ändern. Für den Tooltiptext gibt es ebenfalls einen Standardtext. Um das Verbindungsicon nutzen zu können, muss die Option [OShowlcon](#) aktiviert sein. Bei Menüs ist zusätzlich [OShowMenu](#) erforderlich.

Siehe auch

[Übersicht Optionen](#) | [OShowlcon](#) | [OShowMenu](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.7 ODialKey

Die Taste zum Anwählen von markiertem Text wird festgelegt.

Eigenschaft ODialKey**Typ**

Long, lesen und schreiben

Bemerkungen

Ist [OHotKey](#) aktiv geschaltet, so kann markierter Text mit einer F-Taste gewählt werden. Der Wert entspricht dem virtuellen Keycode der Taste und kann von 0x70 (F1) bis 0x7B (F12) liegen. Die F-Taste kann mit der Steuertaste und Umschalttaste kombiniert werden.

Siehe auch

[Übersicht Optionen](#) | [OOptions](#) | [ODialSpecialKey](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.8 ODialSpecialKey

Die zusätzliche Steuertaste zum Anwählen von markiertem Text wird festgelegt.

Eigenschaft ODialSpecialKey**Typ**

Long, lesen und schreiben

Bemerkungen

Die Steuertaste zum Anrufen von Nummern in markiertem Text kann noch mit Sondertasten kombiniert werden, die in dieser Eigenschaft festgelegt werden. Mögliche Werte sind:

- 0 = Keine zusätzliche Taste
- 1 = Umschalttaste
- 2 = Steuertaste
- 3 = Umschalt- und Steuertaste

Siehe auch

[Übersicht Optionen](#) | [ODialKey](#) | [OHotKey](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.9 ODropKey

Die Taste zum Auflegen von Gesprächen wird festgelegt.

Eigenschaft ODropKey**Typ**

Long, lesen und schreiben

Bemerkungen

Ist [OHotKey](#) aktiv geschaltet, so kann ein Gespräch mit einer F-Taste aufgelegt werden. Der Wert entspricht dem virtuellen Keycode der Taste und kann von 0x70 (F1) bis 0x7B (F12) liegen. Die F-Taste kann mit der Steuertaste und Umschalttaste kombiniert werden. Die Werte sollten sich von [ODialKey](#) unterscheiden.

Siehe auch

[Übersicht Optionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.10 ODropSpecialKey

Die zusätzliche Steuertaste zum Auflegen von Gesprächen wird festgelegt.

Eigenschaft ODropSpecialKey**Typ**

Long, lesen und schreiben

Bemerkungen

Die Steuertaste zum Auflegen von Gesprächen kann noch mit Sondertasten kombiniert werden, die in dieser Eigenschaft festgelegt werden. Mögliche Werte sind:

- 0 = Keine zusätzliche Taste
- 1 = Umschalttaste
- 2 = Steuertaste
- 3 = Umschalt- und Steuertaste

Siehe auch

[Übersicht Optionen](#) | [ODropKey](#) | [OHotKey](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.11 OHotKey

Schaltet die Schnellwahl über Tastaturcodes zu.

Eigenschaft OHotKey**Typ**

Bool, lesen und schreiben

Bemerkungen

Legt fest, ob markierter Text mit einer Tastenkombination angewählt werden kann. Das kann eine der Tasten F1 bis F12 sein, Zusätzlich können die Umschalt- und / oder die Strg-Taste gedrückt sein. Befindet sich eine wählbare Nummer im Text, so wird diese angewählt. Die Schnellwahl- und Zusatztasten werden mit [ODialKey](#) und [ODialSpecialKey](#) festgelegt.

Voreinstellung

False

Siehe auch

[Übersicht Optionen](#) | [OOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.12 OIniFileName

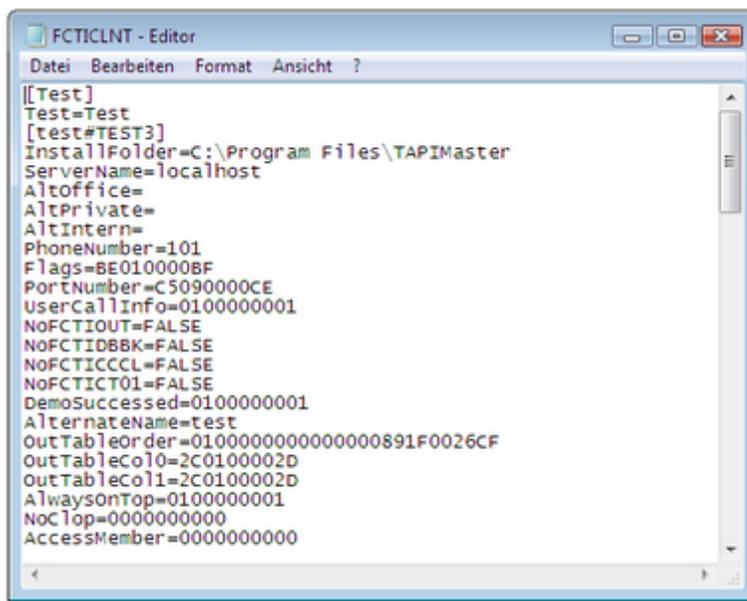
Der Name der Konfigurationsdatei wird zurückgeliefert.

Eigenschaft OIniFileName**Typ**

String, nur lesen

Bemerkungen

Die CTI-Schnittstelle speichert ihre Einstellungen in einer Konfigurationsdatei. Wenn Sie als Programmierer für die Einstellungen Ihres Clients keine separate Datei anlegen wollen, können Sie diese verwenden. Die Datei liegt im Windows-INI Format vor. Der Sektionsname ergibt sich aus dem Benutzernamen + # + Computernamen. Beispiel: [mueller#wk12]. Werfen Sie einen Kurzen Blick in eine der vorhandenen INI-Dateien, um keine vorhandenen Schlüsselwörter zu benutzen.



```
[Test]
Test=Test
[test#TEST3]
InstallFolder=C:\Program Files\TAPIMaster
ServerName=localhost
AltOffice=
AltPrivate=
AltIntern=
PhoneNumber=101
Flags=BE010000BF
PortNumber=C5090000CE
UserCallInfo=0100000001
NoFCTIOUT=FALSE
NoFCTIDBBK=FALSE
NoFCTICCL=FALSE
NoFCTICT01=FALSE
DemoSucceeded=0100000001
AlternateName=test
OutTableOrder=010000000000000891F0026CF
OutTableCol0=2C0100002D
OutTableCol1=2C0100002D
AlwaysOnTop=0100000001
NoClop=0000000000
AccessMember=0000000000
```

Siehe auch

[Übersicht Optionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.13 OOptions

Die Optionen werden als Bitmuster bearbeitet.

Eigenschaft OOptions**Typ**

Long, lesen und schreiben

Bemerkungen

Die [Verbindungsoptionen](#) können auch komplett als Bitmuster gelesen und geschrieben werden.

Voreinstellung

0x0000019E

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Optionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.14 OOwnerName

Der Anmeldename wird festgelegt.

Eigenschaft OOwnerName**Typ**

String, lesen und schreiben

Bemerkungen

Die Änderung bewirkt eine Neuansmeldung am Server.

Siehe auch

[Übersicht Optionen](#) | [NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.15 OOwnerNumber

Die eigene Durchwahl wird festgelegt.

Eigenschaft OOwnerNumber**Typ**

String, lesen und schreiben

Bemerkungen

Eigene Durchwahl, welcher der Server benötigt um die Leitung zuzuordnen.

Siehe auch

[Übersicht Optionen](#) | [NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.16 OPhoneIcon

Ein Telefonsymbol wird bei eingehenden Anrufen eingeblendet.

Eigenschaft OPhoneIcon**Typ**

Bool, lesen und schreiben

Bemerkungen

Bei eingehenden Anrufen kann in der Taskleiste ein kleines Telefonsymbol eingeblendet werden, analog dem Briefsymbol in MS Outlook. Im Tooltip des Icons wird dann die Rufnummer sofern vorhanden angezeigt. Das Icon verfügt über ein kleines Menü, mit dem der Anrufer zurückgerufen werden kann. Das Icon wird mit [TRemovePhoneIcon](#) entfernt. Ein Doppelklick auf das Icon löst den Event [OnShowPhoneList](#) aus, hier sollten Sie die Anruferliste sofern vorhanden anzeigen.

**Beispiel**

[Mini CTI Client \(Delphi\)](#)

Voreinstellung

True

Siehe auch

[Übersicht Optionen](#) | [OOptions](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.17 OShow Icon

Ein Verbindungssymbol wird angezeigt.

Eigenschaft OShowIcon**Typ**

Bool, lesen und schreiben

Bemerkungen

In der Taskleiste wird ein Verbindungssymbol angezeigt, welches den Status der Netzwerkverbindung erkennen lässt. Es kann mit einem benutzerdefinierten Menü und Tooltip versehen werden. Die Icons für stehende unterbrochene Verbindung können durch eigene Icons ausgetauscht werden.

**Voreinstellung**

True

Beispiel

[Mini CTI Client \(Delphi\)](#)

Siehe auch

[Übersicht Optionen](#) | [OOptions](#) | [OConnectionIconOptions](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.18 OShow Menu

Das Verbindungsicon wird mit einem Menü versehen.

Eigenschaft OShowMenu

Typ

Bool, lesen und schreiben

Bemerkungen

Dem Verbindungsicon in der Taskleiste lässt sich ein Menü zuordnen. Es erscheint dann wenn der Benutzer mit der rechten Maustaste auf das Icon klickt. Mit [OConnectionIconOptions](#) wird das Menühandle übergeben. Die Events des Menüs werden mit OnMenuCommand geliefert. [OShowIcon](#) muss True sein.

Voreinstellung

False

Siehe auch

[Übersicht Optionen](#) | [OOptions](#) | [OConnectionIconOptions](#) | [OShowIcon](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.2.19 OUserCallInfo

Die Art der Benachrichtigung bei eingehenden Anrufen wird bestimmt.

Eigenschaft OUserCallInfo**Typ**

Long, lesen und schreiben

Bemerkungen

Legt fest wie der an der Schnittstelle angedockte Client über einen eingehenden Anruf informiert werden soll. Mögliche Werte sind:

- 0 Keine Information
- 1 Der Telefonclient erhält ein Event, worauf er sich nach vorne bringen soll
- 2 Kundenspezifische Anzeige

Die Kundenspezifische Anzeige wird im Standardclient verwendet.

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Optionen](#) | [OnCallNumber](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3 Telefoniefunktionen

Verwendung der Telefoniefunktionen

Die TAPIMaster®-Schnittstelle kapselt die TAPI-Schnittstelle und reduziert deren Befehlsumfang deutlich. Die auf dieser Seite aufgeführten Funktionen dienen der Steuerung des Telefons oder sind verwandter Natur. Nicht alle Funktionen sind auf allen Anlagen möglich, dies hängt von der Leistungsfähigkeit des jeweiligen Treibers ab.

In der TAPIMaster®-Schnittstelle hat jeder Anruf eine eindeutige Nummer (Hat nichts mit der Rufnummer zu tun), mit der die Anrufe unterschieden werden können. Sie sollten sich diese Nummern intern merken, da sie bei vielen Telefoniebefehlen gebraucht werden. Sie wird als Anruf-ID bezeichnet.

Präfix

Die Telefoniefunktionen haben den Präfix "T" vorangestellt.

Single Call Funktionen

TMakeCall	Tätigt einen neuen Anruf.
TCallAnswer	Nimmt einen Anruf entgegen.
TCallDrop	Legt einen Anruf auf.

Multicall Funktionen

TAddCallToConference	Fügt einen bestehenden Anruf einer Konferenz hinzu.
TAddNumberToConference	Ruft einen weiteren Teilnehmer an und holt diesen in die Konferenz.
TCallChange	Veraltet. Benutzen Sie TCallToggle .
TCallHold	Ein Anruf wird gehalten.
TCallRedirect	Ein Anruf wird weitergereicht.
TCallRedirection	Veraltet. Benutzen Sie TCallRedirect .
TCallToggle	Schaltet zwischen aktiven und gehaltenem Anruf hin und her.
TCallTransfer	Zwei Anrufe werden verbunden.
TCallUnHold	Holt einen gehaltenen Anruf zurück.
TPickup	Holt einen Anruf von einem anderen Teilnehmer heran.
TProxyMakeCall	Ein anderes Telefon ruft Teilnehmer an, tätigt Rückfrage zum Ziel und verbindet.

Hilfsfunktionen

TLineReset	TAPI-Leitung wird zurückgesetzt.
TReDial	Startet eine Wahlwiederholung.
TRemovePhonenumber	Das Anrufsymbol wird entfernt.

TSendCallReminder	Eine Rückruferinnerung wird gesendet.
TSendRecall	Veraltet. Verwenden Sie TSendCallReminder .
TSetForwarding	Die Rufumleitung wird zu- oder abgeschaltet.
TSetRedirection	Veraltet. Benutzen Sie TSetForwarding .

Parameter

TForwardingState	Der Status der Rufumleitung wird abgefragt oder geändert.
TRedirection	Veraltet. Benutzen Sie TForwardingState .
TForwardingDest	Die Zielnummer der Rufumleitung wird abgefragt oder geändert.
TRedirectionNumber	Veraltet. Benutzen Sie TForwardingDest .

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.1 TAddCallToConference

Fügt ein Gespräch zu einer Konferenz hinzu.

**Funktion TAddCallToConference(
Long ICallID
)**

Parameter

ICallID

Anruf-ID des Gesprächs, welches zu einer Konferenz hinzugefügt werden soll. Wenn hier 0 steht dann versucht die CTI-Engine automatisch entweder eine Konferenz aufzubauen oder den wahrscheinlichsten Teilnehmer hinzuzufügen.

Bemerkungen

Mit dieser Funktion können Sie bereits bestehende Gespräche zu einer Konferenz hinzufügen. In der Regel werden Sie ein gehaltenes Gespräch nach einer Rückfrage mit in das aktive Gespräch holen. Nicht alle Anlagen unterstützen diese Funktion. Insbesondere Konferenzen mit mehr als drei Teilnehmern werden nur von wenigen Herstellern unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TAddNumberToConference](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.2 TAddNumberToConference

Tätigt einen Anruf und holt diesen in die Konferenz.

```
Funktion TAddNumberToConference(  
    String bstrNumber  
)
```

Parameter

bstrNumber

Rufnummer des Teilnehmers, welcher angerufen und in die Konferenz geholt werden soll.

Bemerkungen

Die Funktion ruft *bstrNumber* an. Nimmt dieser das Gespräch entgegen, so wird er in die Konferenz mit einbezogen bzw. das aktive Gespräch zu einer Dreierkonferenz ausgebaut. Nicht alle Anlagen unterstützen diese Funktion. Insbesondere Konferenzen mit mehr als drei Teilnehmern werden nur von wenigen Herstellern unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TAddCallToConference](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.3 TCallAnswer

Nimmt ein Gespräch entgegen.

```
Funktion TCallAnswer(  
    Long ICallID  
)
```

Parameter

ICallID

Anruf-ID des Gesprächs, das entgegengenommen werden soll. Wird 0 übergeben, so wird versucht den ersten Anruf in der eigenen Warteschlange entgegen zunehmen, der sich im State [OFFERING](#) befindet.

Bemerkungen

Nimmt einen eingehenden Anruf entgegen. Die Funktion hat nur dann Erfolg, wenn die Telefone ins Freisprechen schalten können, die meisten ISDN- und Systemtelefone können dies.

Beispiel

[Mini CTI Client \(Delphi\)](#)

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.4 TCallDrop

Ein Anruf wird aufgehlegt.

```
Funktion TCallDrop(  
    Long ICallID  
);
```

Parameter

ICallID

Anruf-ID des Gesprächs, welches aufgehlegt werden soll. Wird 0 übergeben, so wird das erste Gespräch in der Leitung aufgehlegt.

Bemerkungen

Legt ein Gespräch auf. Ist in fast allen Treibern implementiert.

Beispiel

[Mini CTI Client \(Delphi\)](#)

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TMakeCall](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.5 TCallHold

Ein Gespräch wird gehalten.

```
Funktion TCallHold(  
    Long ICallID  
)
```

Parameter

ICallID

Anruf-ID des Gesprächs, das ins Halten gestellt werden soll. Wenn diese 0 ist dann wird der wahrscheinlichste Anruf gehalten.

Bemerkungen

Stellt einen aktiven Anruf in den gehaltenen Zustand. Der Anrufer hört Wartemusik. Mit [TCallUnHold](#) kann der Anruf zurückgeholt werden. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TCallToggle](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.6 TCallRedirect

Ein Gespräch wird weitergeleitet.

```
Funktion TCallRedirect(  
    Long ICallID  
    Long ICallID2  
)
```

```
Long ICallID  
String bstrNumber  
)
```

Parameter*ICallID*

Anruf-ID des Gesprächs, das umgeleitet werden soll.

bstrNumber

Zielnummer, zu der ein Anruf umgeleitet werden soll.

Bemerkungen

Die Funktion leitet Anrufe zu anderen Teilnehmern weiter. Es ist gleichgültig, ob der Anruf bereits verbunden ist oder noch nicht entgegengenommen wurde. Bei guten Anlagen können auch anklopfende Gespräche weitergeleitet werden. Die Funktion wird nicht von allen Telefonanlagen unterstützt. Wenn ICallID Null ist dann wird der wahrscheinlichste Anruf im State OFFERING / ACCEPTED weitergeleitet. Ist kein Anruf in dem State dann ist das wahrscheinlichste verbundene Gespräch Gegenstand der Weiterleitung.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.7 TCallToggle

Wechselt zwischen zwei Anrufen hin und her.

```
Funktion TCallToggle(  
    Long IOldCall,  
    Long INewCall  
);
```

Parameter*IOldCall*

Anruf-ID des Gesprächs, das gerade verbunden ist und ins Halten gestellt werden soll.

INewCall

Anruf-ID des Gesprächs, welches gerade gehalten wird und verbunden werden soll.

Bemerkungen

Die Funktion wechselt zwischen zwei Anrufen hin und her. Ein Gespräch ist jeweils verbunden, das andere Gespräch wird gehalten. Die Parameter können auch vertauscht sein, die Funktion ist fehlertolerant. Sind die Parameter 0 dann Sucht die CTI-Engine automatisch nach den beiden zu tauschenden Anrufen. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TCallHold](#) | [TCallUnHold](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.8 TCallTransfer

Zwei Anrufe werden verbunden.

```
Funktion TCallTransfer(  
    Long ICall1,  
    Long ICall2  
)
```

Parameter

ICall1

Anruf-ID des ersten Anrufs

ICall2

Anruf-ID des zweiten Anrufs

Bemerkungen

Zwei Anrufe sollen miteinander verbunden werden. Meist ist ein Gespräch davon aktiv und das andere gehalten (hört Wartemusik). Die Reihenfolge in der die Anruf-IDs übergeben werden spielt keine Rolle. Die Funktion wird nicht von allen Telefonanlagen unterstützt. Ist einer oder beide Parameter 0 dann sucht die CTI-Engine automatisch nach den beiden Anrufen und verbindet die mit der größten Wahrscheinlichkeit.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.9 TCallUnHold

Ein gehaltenes Gespräch wird zurückgeholt.

```
Funktion TCallUnHold(  
    Long ICallID  
)
```

Parameter

ICallID

Anruf-ID des Gesprächs, welches wieder verbunden werden soll. Ist der Parameter 0 dann wird das letzte gehaltene Gespräch zurückgeholt.

Bemerkungen

Ein mit [TCallHold](#) gehaltener Anruf kann zurückgeholt werden, so dass das Gespräch wieder aktiv ist. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TCallToggle](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.10 TForwardingDest

Das Rufumleitungsziel wird bearbeitet.

Eigenschaft TForwardingDest**Typ**

String, lesen und schreiben

Bemerkungen

Das Rufumleitungsziel kann abgefragt werden. Wird die Eigenschaft geändert dann wird eine neue Rufumleitung mit dem geänderten Ziel aktiv.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TForwardingState](#) | [TSetForwarding](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.11 TForwardingState

Der Status der Rufumleitung wird bearbeitet.

Eigenschaft TForwardingState**Typ**

Bool, lesen und schreiben

Bemerkungen

Schaltet die Rufumleitung zu oder ab ohne das Rufumleitungsziel zu verändern oder liefert den Status der Rufumleitung zurück.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TForwardingDest](#) | [TSetForwarding](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.12 TLineReset

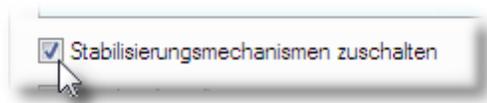
Die TAPI-Leitung wird zurückgesetzt.

Funktion TLineReset()**Parameter**

Keine

Bemerkungen

Mitunter kann es vorkommen, dass der TAPI-Treiber nicht mehr reagiert. Diese Funktion setzt die Leitung zurück, in den meisten Fällen sollte dann das Problem behoben sein. Sie haben auch die Möglichkeit, im CTI-Server eine Autokorrektur zuzuschalten, welche versucht solche Probleme zu beheben. In der Serverkonsole finden Sie auf der Seite "TAPI-Sonderbehandlung" dazu die Möglichkeit.

**Siehe auch**

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.13 TMakeCall

Ein Anruf wird gestartet.

```
Funktion TMakeCall(  
    String bstrNumber  
);
```

Parameter

bstrNumber

Rufnummer der angerufenen Stelle

Bemerkungen

Sollte bereits ein aktives Gespräch bestehen, wird versucht dieses ins Halten zu stellen und einen neuen Anruf zu tätigen. Sie können diese Funktion auch für Rückfragen einsetzen.

Beispiel

[Mini CTI Client \(Delphi\)](#)

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TCallDrop](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.14 TPickup

Ein eingehendes, nicht entgegengenommenes Gespräch wird herangeholt.

```
Funktion TPickup(  
    String bstrNumber  
)
```

Parameter

bstrNumber

Durchwahl des Apparats, von dem ein Gespräch herangeholt werden soll.

Bemerkungen

Mit Hilfe dieser Funktion lassen sich Gespräche von anderen Apparaten heran holen. TAPIMaster® arbeitet hier teilweise mit Ersatzfunktionen gegenüber der TAPI, so dass auch ein anlagenübergreifendes Heranholen von Gesprächen möglich ist. Sollte die Funktion trotzdem

fehlschlagen, kann TAPIMaster® direkt Hardware-Steuerbefehle absetzen um das Gespräch heranzuholen. Je nach Anlage können auch anklopfende Gespräche gepickt werden.

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.15 TProxyMakeCall

Ein Anruf wird über ein Proxytelefon gestartet. Nützlich bei einigen DECT Telefonen.

```
Funktion TProxyMakeCall(  
    String bstrNumber,  
    String bstrProxy,  
);
```

Parameter

bstrNumber

Rufnummer der angerufenen Stelle

bstrProxy

Durchwahl des vermittelnden Telefons, muss ein Systemtelefon sein.

Bemerkungen

Ein Teilnehmer kann nicht selbst über TAPI telefonieren. Ein Vermittlertelefon ruft den Akteur an. Der Akteur hebt ab. Daraufhin startet das Vermittlertelefon eine Rückfrage zum eigentlichen Anrufziel. Das Gespräch wird noch in der Anwahl übergeben.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TCallDrop](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.16 TReDial

Die Wahlwiederholung wird gestartet.

```
Funktion TReDial()
```

Parameter

Keine

Bemerkungen

Tätigt eine Wahlwiederholung der zuletzt angerufenen Nummer.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.17 TRemovePhoneIcon

Das Telefonsymbol wird entfernt.

Funktion TRemovePhoneIcon()

Parameter

Keine

Bemerkungen

Bei eingehenden Anrufen kann ein kleines Telefonsymbol eingeblendet sein, wenn die Option [CTICLIENT_PHONEICON](#) aktiviert wurde. Mit dieser Funktion kann es wieder entfernt werden wenn das Telefonat bearbeitet ist.



Siehe auch

[Übersicht Telefoniefunktionen](#) | [OOptions](#)

[Send feedback to TAPIMaster®](#)
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.18 TSendCallReminder

Eine Anruferinnerung wird gesendet.

```
Funktion TSendCallReminder(  
    String bstrOther  
    String bstrNumber  
    String bstrMessage  
)
```

Parameter

bstrOther

Durchwahl des Teilnehmers, welcher informiert werden soll.

bstrNumber

Rufnummer des Anrufers, der zurückgerufen werden soll.

bstrMessage

Kurze Nachricht (bis 255 Zeichen), die das Problem des Anrufers beschreiben.

Bemerkungen

Ein Anruf kommt herein. Es stellt sich heraus dass der Anrufer einen Kollegen sprechen wollte der aber gerade nicht da ist. Mit dieser Funktion kann der Kollege schnell und einfach über den Anruf informiert werden. Es wird ein Event vom Typ [OnCallReminder](#) im Client des Kollegen ausgelöst. Ist

dieser nicht angemeldet, erhält er den Event bei seiner Anmeldung. Somit gehen weniger Informationen verloren, die Daten können auf dem Zielclient gleich für eine Anrufvorlage genutzt werden, zeitraubendes Abschicken von Mails entfällt.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.3.19 TSetForwarding

Die Rufumleitung wird zu- oder abgeschaltet.

```
Funktion TSetForwarding(  
    Bool bSet,  
    String bstrNumber  
)
```

Parameter

bSet

Zu- oder Abschalten der Umleitung

bstrNumber

Nummer, zu der umgeleitet werden soll.

Bemerkungen

Schaltet eine Rufumleitung oder zu oder ab. Als Antwort darauf erhält der Teilnehmer einen Event vom Typ [OnForwardingState](#). Die Funktion kann fehlschlagen, wenn die Zielnummer bereits umgeleitet hat, die Rechte zur Umleitung nicht vorliegen oder die TAPI-Schnittstelle der Anlage die Funktion nicht unterstützt. Im letzten Fall gibt es die Möglichkeit, die Rufumleitung über Telefoncodes zu steuern, diese werden in der Serverkonsole auf der Seite "TAPI-Sonderbehandlungen" eingestellt.

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Telefoniefunktionen](#) | [TForwardingState](#) | [TForwardingDest](#)

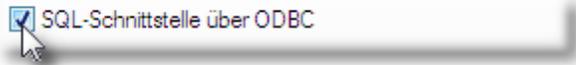
[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.4 SQL

Verwendung der SQL-Funktionen

In Telefonieanwendungen kann es erforderlich sein, auf Datenbankbestände zuzugreifen um z. B. zu einer Rufnummer weitere Daten wie Adresse, Notizen usw. anzeigen zu können. Der CTI-Server unterhält bereits Datenbankverbindungen um Rufnummern identifizieren zu können oder Gesprächsdatensätze zu speichern. Mit einer dritten Datenbankverbindung haben Sie die Möglichkeit, individuelle Abfragen zu tätigen. Richten Sie dazu erst die Datenquelle auf dem CTI-Server ein. Auf der Seite "Datenbankschnittstelle" der Serverkonsole finden Sie die entsprechenden Einstelloptionen.



Danach empfiehlt es sich die Befehle [SQLLogin](#), [SQLBegin](#), [SQLNext](#) und [SQLEnd](#) zuerst einmal mit dem [CTI-Browser](#) zu testen.

Präfix

Die SQL-Funktionen haben den Präfix "SQL" vorangestellt.

Adressen aus der Stammdatenbank holen

Aus der Datenbank für die Rufnummernidentifikation lassen sich auch umgekehrt Rufnummern zu Namen ermitteln.

SQLGetDbBook	Holt einzelne Datensätze oder eine komplette Adreßliste aus der Stammdatenbank.
------------------------------	---

Abfragen mit zusätzlicher Datenbankverbindung

Nutzen Sie diese Befehle für individuelle Datenbankabfragen.

SQLBegin	Meldet Client bei der Datenbank an.
SQLEnd	Startet eine Datenbankabfrage.
SQLLogin	Setzt eine Abfrage fort.
SQLNext	Beendet eine Abfrage.

Möglicher Ablauf einer Sitzung

1. Anmelden an Datenbank mit [SQLLogin](#)("Paßwort");
2. [OnSQLLogin](#) liefert True zurück, Anmeldung erfolgreich
3. 50 Datensätze anfordern mit [SQLBegin](#)("SELECT * FROM members",50); Funktion liefert Request von 1 zurück.
4. Datensätze kommen mit Event [OnSQLData](#) an.
5. Weitere 30 Datensätze werden angefordert für Abfrage Nummer 1: [SQLNext](#)(1,30);
6. Datensätze kommen mit Event [OnSQLData](#) an.
7. Abfrage Nummer 1 beenden mit [SQLEnd](#)(1);
8. Das Event [OnSQLEnd](#) bestätigt das Ende der Abfrage.

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.4.1 SQLBegin

Eine SQL-Abfrage wird gestartet.

```
Funktion SQLBegin(  
    String bstrCommand,  
    ULong dwCount,  
    ULongPtr pdwRequest  
)
```

Parameter

bstrCommand

SQL-Kommando, welches an die Datenbank geschickt wird.

dwCount

Maximale Anzahl der Datensätze, die abgefragt werden soll.

pdwRequest

Abfrage-ID als Rückgabewert

Bemerkungen

Die Funktion startet eine Datenbankabfrage. Es kann ein beliebiger SQL-String übergeben werden, sofern der Administrator hier keine Beschränkungen vorgenommen hat. Da die Zahl der ermittelten Datensätze sehr groß werden kann sollte sie mit *dwMaxCount* sinnvoll beschränkt werden. Die Abfrage lässt sich mit [SQLNext](#) fortsetzen und [SQLEnd](#) beenden. Mit **SQLBegin** gestartete Abfragen werden nach einigen Minuten selbständig auf dem Server terminiert. Bei jeder Abfrage wird eine Abfrage-ID zurückgeliefert. Diese wird beim Aufruf von [SQLNext](#) und [SQLEnd](#) benötigt. Wird 0 zurückgeliefert so konnte die Abfrage nicht gestartet werden. Das ist der Fall wenn der Benutzer nicht an der Datenbank angemeldet ist.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [OnSQLData](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.4.2 SQLEnd

Eine SQL-Abfrage wird beendet.

```
Funktion SQLEnd(  
    ULong dwRequest  
)
```

Parameter

dwRequest

Abfrage-ID

Bemerkungen

Eine Abfrage wird beendet. Daraufhin erhält der Client einen Event vom Typ [OnSQLEnd](#). Da für jede Abfrage ein Statementhandle im Datenbankzugriff angelegt wird, sollten nicht benötigte Abfragen so schnell wie möglich terminiert werden. Beim Aufruf von **SQLEnd** übergeben Sie die Abfrage-ID, welche bei [SQLBegin](#) zurückgeliefert wurde. Nicht geschlossene Abfragen werden nach einigen Minuten automatisch beendet.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [OnSQLEnd](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.4.3 SQLGetDbBook

Rufnummerdatensätze zu Namen oder Namensbestandteilen werden gesucht.

```
Funktion SQLGetDbBook(  
    String bstrText  
)
```

Parameter

bstrText

Suchkriterium

Bemerkungen

Die Funktion sucht in der Datenbank für die Rufnummernidentifikation nach Rufnummern. Hier sind Namen oder Namensfragmente gegeben und Datensätze mit Namen und Rufnummern werden zurückgeliefert. Ein "Sch" zum Beispiel sucht nach allen Namen wie Schmidt oder Schulz. Wird ein Leerstring übergeben, so erstellt die Datenbank ein komplettes Telefonbuch und schickt es an den Client zurück. Die zurückgesendeten Daten lösen im Client einen Event vom Typ [OnDbBookRecord](#) aus. Bevor diese Funktion genutzt werden kann ist es erforderlich dass die Datenbankschnittstelle des CTI-Servers eingerichtet wurde. Die Rufnummernerkennung auf der Seite "Datenbankschnittstelle" der Serverkonsole sollte aktiviert und eingerichtet sein.



Siehe auch

[Übersicht SQL](#) | [OnDbBookRecord](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.4.4 SQLLogin

Die Anmeldung an die Datenbank erfolgt.

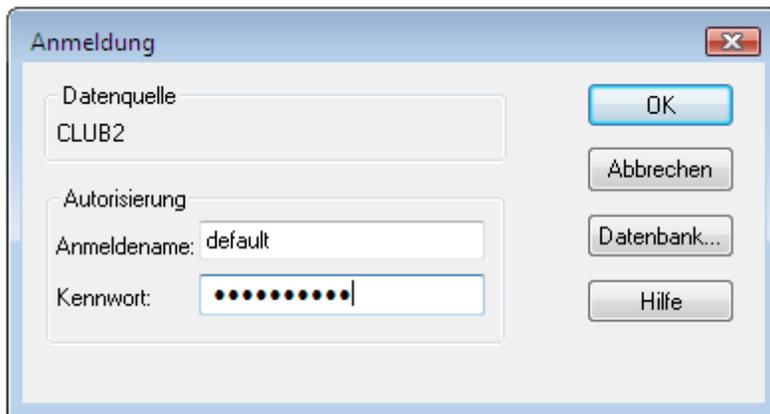
```
Funktion SQLLogin(  
    String bstrPwd  
)
```

Parameter

bstrPwd
Paßwort der Datenbank

Bemerkungen

Um die SQL-Schnittstelle nutzen zu können, meldet sich der Client mit dem Paßwort der Datenbank an. Das Paßwort wurde vom Administrator beim Einrichten der ODBC-Datenquelle vergeben.



bstrPwd kann auch ein leerer String sein, wenn kein Paßwort vergeben wurde. Als Antwort darauf erhält der Client ein Event vom Typ [OnSQLLogin](#).

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [OnSQLLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.4.5 SQLNext

SQL-Abfrage wird fortgesetzt.

```
Funktion SQLNext(  
    ULong dwRequest,  
    ULong dwCount,  
    BoolPtr pbSuccess  
)
```

Parameter

dwRequest
Abfrage-ID

dwCount
Maximale Anzahl von Datensätzen

pbSuccess
Rückgabewert, True bei Erfolg

Bemerkungen

Eine mit [SQLBegin](#) gestartete Abfrage wird fortgesetzt. Dazu wird die Abfrage-ID übergeben, welche [SQLBegin](#) zurückgeliefert hat. Die Anzahl der maximal zu liefernden Datensätze wird begrenzt auf so viele Datensätze, wie in nächster Zeit sinnvoll weiterverarbeitet werden kann. Die Lebensdauer einer Abfrage verlängert sich durch jedes **SQLNext**. Ist der Client nicht an die Datenbank angemeldet, so schlägt die Funktion fehl und liefert False zurück.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [OnSQLData](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5 Zusatzfunktionen

Verwendung der Zusatzfunktionen

Die CTI-Schnittstelle wurde um einige nützliche Hilfsfunktionen ergänzt, welche zwar meist nichts mit Telefonie zu tun haben aber die Programme bereichern können. So wird Chat und das Senden von Nachrichten unterstützt.

Präfix

Die Zusatzfunktionen haben den Präfix "E" vorangestellt.

Nachrichten senden

EChatStart	Startet eine Chatsitzung.
EChatMessage	Die Chatnachricht wird übertragen.
EChatEnd	Die aktive Chatsitzung wird beendet.
ESendUserToUser	Eine Kurznachricht wird geschickt.
EUserSendToDataBase	Sendet benutzerdefinierte Nachrichten.

Sonstige

EExpandInterNumber	Expandiert eine Durchwahl.
EGetDateTime	Wandelt time_t in Standardstring um.
EGetFormatDateTime	Wandelt time_t in individuell formatierten String um.
ESetAcdReady	Bereitschaft zum Gesprächsempfang in der ACD-Gruppe signalisieren.
ESetMainWindow	Übergibt das Hauptfensterhandle an die Schnittstelle.
EShowInfoDialog	Zeigt Versionsinfo der Schnittstelle an.
EConvertToCanonical	Wandelt eine Rufnummer in das kanonische Format um.

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.1 EChatEnd

Die aktive Chatsitzung wird beendet.

Funktion EChatEnd()**Parameter**

Keine

Bemerkungen

Jeder der beiden Chatpartner kann diese Funktion aufrufen. Darauf hin erhalten beide Seiten einen Event vom Typ [OnChatStop](#). Die Anwendung kann dann die Chatfenster schließen oder deaktivieren.

Siehe auch

[Übersicht Zusatzfunktionen](#) | [EChatStart](#) | [EChatMessage](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.2 EChatMessage

Die Chatnachricht wird übertragen.

```
Funktion EChatMessage(  
    String bstrMessage  
)
```

Parameter

bstrMessage

Nachrichtentext, der in der Chatbox steht.

Bemerkungen

Es wird der komplette Text in der Chatbox übertragen. Die Funktion sollte aufgerufen werden sobald sich im jeweiligen Eingabefeld etwas ändert. Die Funktion ruft beim Chatpartner den Event [OnChatMessage](#) auf.

Siehe auch

[Übersicht Zusatzfunktionen](#) | [EChatStart](#) | [EChatEnd](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.3 EChatStart

Startet eine Chatsitzung.

```
Funktion EChatStart(  
    String bstrDest  
)
```

Parameter

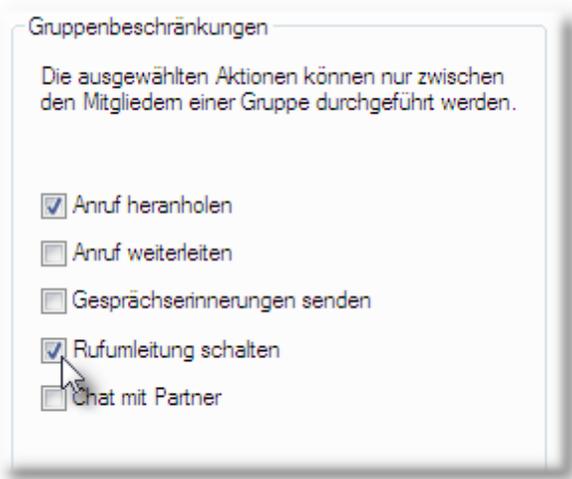
bstrDest

Durchwahl des Chatpartners

Bemerkungen

Der Chatpartner muss angemeldet und die Rechte zum Chat vorhanden sein; es darf seinerseits noch keine Chatsitzung geben. Wenn die Anmeldung zum Chat zulässig war, erhalten beide Partner einen Event vom Typ [OnChatStart](#). Daraufhin kann die Anwendung Ein- und Ausgabefelder für die Chatsitzung einblenden. Sollte Chat nicht funktionieren könnte es daran liegen dass Chat auf die

Mitglieder der gleichen Gruppe beschränkt ist. Auf der Seite "Gruppenverwaltung" der Serverkonsole können Sie diese Beschränkungen aufheben.



Siehe auch

[Übersicht Zusatzfunktionen](#) | [EChatMessage](#) | [EChatEnd](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.4 EConvertToCanonical

Umwandlung von Rufnummern in das kanonische Format.

```
Funktion EConvertToCanonical(  
    String bstrInput,  
    StringPtr pbstrOutput  
)
```

Parameter

bstrInput

Rufnummer, die in das kanonische Format umgewandelt werden soll.

pbstrOutput

Umgewandelte Rufnummer

Bemerkungen

Sie können diese Funktion benutzen um Datenbestände mit Rufnummern in das kanonische Format umzuwandeln. Der Ausgabestring sollte mindestens 32 Zeichen groß sein.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.5 EExpandInternNumber

Expandiert eine Durchwahl.

```
Funktion EExpandInternNumber(  

String bstrNumber,  

StringPtr pbstrReturn  

)
```

Parameter

bstrNumber

Interne Rufnummer, welche erweitert werden soll.

pbstrReturn

Rückwert, der die expandierte Rufnummer enthält.

Bemerkungen

Erweitert eine Rufnummer um die Landeskenning, Vorwahl und ggf. Anlagenrufnummer.

Beispiel

Bei diesen Einstellungen liefert EExpandInternNumber("22") in Deutschland den Wert +49 (6033) 44422.

Amt	Int...	Filter ...	Filter ...	Vorwahl	Anlagennummer
...	0	0	0	6033	444

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.6 EGetDateTime

Wandelt time_t in Standardstring um.

```
Function EGetDateTime(  

Long ITime,  

StringPtr pbstrTimeString  

)
```

Parameter

ITime

Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.

pbstrTimeString

Nullterminierter String in einem Standardformat: 2007-01-12T00:53:32.875

Bemerkungen

Nutzen Sie diese Funktion wenn Sie ein gut sortierbares Format brauchen. Viele Programme wie Excel nutzen dieses Format.

Siehe auch[Übersicht Zusatzfunktionen](#)[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.7 EGetFormatDateTime

Wandelt time_t in individuell formatierten String um.

```

Function EGetFormatDateTime(
    Long lTime,
    String bstrFormat,
    StringPtr pbstrTimeString
)

```

Parameter*lTime*Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.*bstrFormat*

String mit Datumsformat. Dieses kann aus den folgenden Formatzeichen gebildet werden.

Zeichen	Erklärung
%a	Abgekürzter Wochentag
%A	Ausgeschriebener Wochentag
%b	Abgekürzter Monatsname
%B	Ausgeschriebener Monatsname
%c	Lokales Datum und Uhrzeit, abhängig von Systemeinstellungen
%d	Monatstag (01 – 31)
%H	Stunde im 24Stunden-Format (00 – 23)
%I	Stunde im 12Stunden-Format (01 – 12)
%j	Tag im Jahr (001 – 366)
%m	Monat als Zahl (01 – 12)
%M	Minute (00 – 59)
%p	Lokaler Indikator für vormittag / nachmittag bei 12stündiger Anzeige
%S	Sekunden (00 – 59)
%U	Kalenderwoche, beginnend mit Sonntag (00 – 53)
%w	Wochentag als Zahl, beginnend mit Sonntag (0 – 6)
%W	Kalenderwoche, beginnend mit Montag (00 – 53)
%x	Lokales Datum, abhängig vom System
%X	Lokale Zeit, abhängig vom System
%y	2stellige Jahresangabe (00 – 99)
%Y	4stellige Jahresangabe

%z,%Z	Zeitzone
%%	Prozentzeichen

pbstrTimeString

Nullterminierter String in einem individuellen Format.

Bemerkungen

Die Formatzeichen können beliebig kombiniert werden.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.8 ESendUserToUser

Eine Kurznachricht wird geschickt.

```
Funktion ESendUserToUser(
    String bstrOther,
    String bstrMessage
)
```

Parameter

bstrOther

Durchwahl des anderen Teilnehmers

bstrMessage

Kurznachricht (max 220 Zeichen), die an ihn geschickt wird.

Bemerkungen

Verschiedene Benutzer können sich mit dieser Funktion Kurznachrichten schicken, ohne den Mailclient bemühen zu müssen. Auf dem Client des Empfängers wird der Event [OnUserToUser](#) ausgelöst.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.9 ESetAcidReady

Der Client der Servicegruppe teilt dem Server seine Bereitschaft zum Gesprächsempfang mit.

```
Funktion ESetAcidReady(
    Bool bReady,
)
```

Parameter

bReady

Ist True wenn der Client für die Hotline zur Verfügung stehen soll.

Bemerkungen

Der Client muss Mitglied der Servicegruppe sein.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.10 ESetMainWindow

Übergibt das Hauptfensterhandle an die Schnittstelle.

```
Funktion ESetMainWindow(  
    ULong hWnd  
)
```

Parameter

hWnd

Fensterhandle der Anwendung

Bemerkungen

Übergibt der Schnittstelle das Handle des Hauptfensters der Anwendung. Wird benötigt, um Unterdialoge der Schnittstelle an der richtigen Position anzuzeigen.

Beispiel

[Mini CTI Client \(Delphi\)](#)

Siehe auch

[Übersicht Zusatzfunktionen](#) | [EShowInfoDialog](#) | [NNetworkConfigDialog](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.11 EShow InfoDialog

Zeigt Versionsinfo der Schnittstelle an.

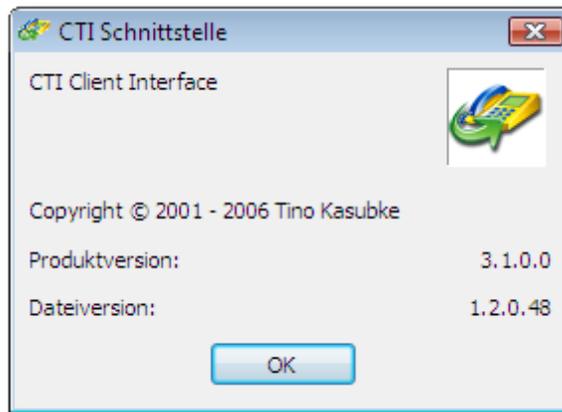
```
Funktion EShowInfoDialog()
```

Parameter

Keine

Bemerkungen

Zeigt einen kleinen Dialog mit Informationen zur Schnittstelle an. Hersteller, Produkt- und Dateiversion der Schnittstelle werden angezeigt.

**Siehe auch**

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.5.12 EUserSendToDataBase

Sendet benutzerdefinierte Nachrichten.

```
Funktion EUserSendToDataBase(  
    Long ICommand,  
    String bstrText  
)
```

Parameter

ICommand

Nummer des Befehls

bstrText

Übermittelte Parameter

Bemerkungen

Mit dieser Funktion können benutzerdefinierte Nachrichten an die Serverschnittstelle geschickt werden. Dort lösen diese einen Event vom Typ [OnUserToDataBase](#) aus. Anhand von *ICommand* werden die dort eingehenden Befehle unterschieden. Die Serverschnittstelle kann auf die Nachricht antworten und löst auf der Clientseite den Event [OnDatabaseToUser](#) aus.

Beispiel

[Excel Client](#)

Siehe auch

[Übersicht Zusatzfunktionen](#) | [ActiveX \(Server\)](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6 Events

Telefonie

OnCallNumber	Rufnummer wird übermittelt.
OnCallState	Zustand des Gesprächs ändert sich.
OnConfNotify	Ein- und Austritt in eine Konferenz
OnIdentity	Rufnummernidentifikation in Serverdatenbank oder Telefonbuch-CD.
OnGetIdentity	Veraltet. Benutzen Sie OnIdentity .
OnGetConf	Veraltet. Benutzen Sie OnConfNotify .
OnLocaleInfo	Einstellungen der Telefonanlage und örtliche Einstellungen
OnGetLocale	Veraltet. Benutzen Sie OnLocaleInfo .
OnCallReminder	Anruferinnerung erhalten
OnGetRecall	Veraltet. Benutzen Sie OnCallReminder .
OnForwardingState	Zustand der Rufumleitung wird mitgeteilt.
OnGetRedirection	Veraltet. Benutzen Sie OnForwardingState .
OnGroupMember	Mitglieder der eigenen Gruppe(n)
OnProtocolData	Gesprächsdatensatz wird gesendet.
OnSendAddress	Adresse, welche Angerufen wurde
OnSendUserState	Zustandsinformationen zu Gruppenmitgliedern
OnObjectOfRedirection	Rufumleitungen, die auf diesen Teilnehmer geschaltet werden
OnTellRedirection	Veraltet, Benutzen Sie OnObjectOfRedirect .

Programmsteuerung

OnActivateClient	Veraltet. Verwenden Sie OnRaiseClient .
OnClientClose	Client soll geschlossen und ggf. wieder neu gestartet werden.
OnConnectionState	Zustand der Netzwerkverbindung ändert sich.
OnInvalidStringFormat	Client und Server kommen aus verschiedenen Produkten.
OnLogin	Erfolg des Einloggens
OnMenuCommand	Menü des Tray Icons wurde angeklickt.
OnRaiseClient	Telefonieanwendung soll in den Vordergrund gebracht werden.
OnShowPhoneList	Anrufsliste soll angezeigt werden.

Nachrichten

OnChatStart	Chatsitzung eröffnen.
OnChatMessage	Chatnachricht senden.
OnChatStop	Chatsitzung beenden.
OnLicenseInfo	Lizenzinformationen werden gesendet.
OnGetLicenseData	Veraltet. Benutzen Sie OnLicenseInfo .
OnServerMessage	Nachricht vom Administrator
OnUserToUser	Nachrichten zwischen den Benutzern

Datenbankzugriff

OnDatabaseToUser	Benutzerdefinierter Befehl zur Serverschnittstelle
OnDbBookRecord	Abfrage nach Rufnummern bei vorliegendem Namen
OnSQLData	SQL-Daten aus Abfrage kommen an.
OnSQLEnd	SQL-Abfrage wurde beendet.
OnSQLLogin	Anmeldung an Datenbank erfolgt.

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.1 OnCallNumber

Eine Rufnummer wird übertragen.

```

Event OnCallNumber(
    Long IType,
    Long ICallID
    String bstrNumber
)

```

Parameter

IType

Art des Anrufs. Folgende Werte sind möglich:

- 1: Primäranruf eingehend
- 2: Anklopfender Anruf bei bestehendem Gespräch
- 3: Ausgehender Anruf

ICallID

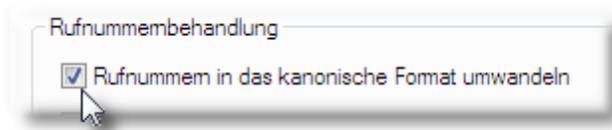
Anruf-ID

bstrNumber

Übertragene Rufnummer, maximal 31 Stellen.

Bemerkungen

Übermittelt Rufnummern, sofern diese mit übertragen wurden. Bei Anrufen aus dem Analognetz und anonymen Anrufen wird keine Rufnummer übertragen. Bei ausgehenden Rufen wird dieser Event ebenfalls ausgelöst. Das Format der Rufnummer kann auf dem Server als kanonisch festgelegt werden (Seite Anlageneinstellungen).



Normalerweise macht ein neuer Anruf auf sich aufmerksam, indem ein Event [OnCallState](#) mit dem Status [LINECALLSTATE_OFFERING](#) (eingehend) oder [LINECALLSTATE_DIALTONE](#) (ausgehend) auftritt. OnCallNumber kann jedoch auch vor allen anderen Events gesendet werden. Die Rufnummer kann sich während des Gesprächs ändern, wenn z. B. jemand dem Benutzer einen Anruf weitervermittelt.

Beispiel

[Mini CTI Client \(Delphi\)](#)
[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Events](#) | [OnCallState](#) | [Gesprächszustände](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.2 OnCallReminder

Eine Anruferinnerung wurde dem Benutzer zugesendet.

```
Event OnCallReminder(  
    String bstrUser,  
    String bstrNumber,  
    String bstrMessage  
)
```

Parameter

bstrUser

Durchwahl des Benutzers, der die Anruferinnerung gesendet hat.

bstrNumber

Nummer, welche zurückgerufen werden soll.

bstrMessage

Zusätzliche Nachricht zum Problem. Hier könnte der Absender erklären, warum *bstrNumber* zurückgerufen werden soll.

Bemerkungen

Die Benutzer können sich gegenseitig Anruferinnerungen schicken. Somit muss man nicht mehr jedem mitteilen, wer dem zurückrufen soll, der Vorgang wird wesentlich kürzer gestaltet. Anruferinnerungen werden mit [TSendCallReminder](#) verschickt.

Siehe auch

[Übersicht Events](#) | [TSendCallReminder](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.3 OnCallState

Übermittelt den Zustand eigener Gespräche.

```
Event OnCallState(  
    Long IState,  
    Long ICallID  
)
```

Parameter

IState

[Gesprächszustand](#), in dem sich der Anruf befindet.

ICallID

Anruf-ID, dessen Zustand sich ändert.

Bemerkungen

Der Zustand eines Gesprächs hat sich geändert. Diese Nachricht wird für die Gespräche des jeweiligen Benutzers übertragen, der das Gespräch führt. Sie sollten eine Liste mit aktiven Anrufen im Programm mitführen. Prüfen Sie bei jedem Aufruf von **OnCallState** ob die Anruf-ID in der Liste enthalten ist und fügen Sie diese bei Bedarf ein. Sollte ein Status [LINECALLSTATE_IDLE](#) gesendet werden, so entfernen Sie den Anruf aus der Liste. In der Liste werden sich meist nur ein oder zwei Anrufe befinden.

Zustände der Gruppenmitglieder werden mit dem Event [OnSendUserState](#) gesendet.

Beispiel

[Mini CTI Client \(Delphi\)](#)

Siehe auch

[Übersicht Events](#) | [OnSendUserState](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.4 OnChatMessage

Eine Chatnachricht wird vom Chatpartner gesendet.

```
Event OnChatMessage(  
    String bstrMessage  
)
```

Parameter

bstrMessage

Nachricht, maximal 255 Zeichen

Bemerkungen

Ein Chat mit einem anderen Teilnehmer besteht und dieser hat uns eine Nachricht geschickt. Zeigen Sie diese Nachricht im Ausgabefenster für diese Nachrichten an. Es wird nicht nur der letzte Buchstabe, sondern der gesamte Text übertragen.

Siehe auch

[Übersicht Events](#) | [EChatMessage](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.5 OnChatStart

Eine Chat-Sitzung wurde eröffnet.

```
Event OnChatStart(  
    Bool bStartOK,  
    String bstrUser  
)
```

Parameter

bStartOK

Zeigt an ob die Sitzung erfolgreich eröffnet wurde.

bstrUser

Durchwahl des anderen Chatpartners

Bemerkungen

Es Gibt zwei Möglichkeiten, dieses Event zu erhalten:

1. Sie haben mit [EChatStart](#) um eine Chatsitzung mit einem Teilnehmer gebeten. Diese kann von den Rechten her nicht erlaubt sein oder der Chatpartner hat bereits eine Sitzung. Dann ist *bStartOK* False. Ansonsten bekommen Sie True zurück, in dem Fall können Sie geeignete Fenster für die Chat Ein- und Ausgabe anzeigen.
2. Jemand möchte eine Chatsitzung mit Ihnen beginnen. Anhand von *bstrUser* sehen Sie wer das ist. Wenn Sie die Chatsitzung ablehnen möchten, so rufen Sie hier [EChatEnd](#) auf. Ansonsten zeigen Sie die Fenster für Ein- und Ausgabe an.

Siehe auch

[Übersicht Events](#) | [EChatStart](#) | [EChatEnd](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.6 OnChatStop

Die Chatsitzung soll beendet werden.

```
Event OnChatStop()
```

Parameter

Keine

Bemerkungen

Die Nachricht wird an beide Chatpartner gesendet, wenn einer von ihnen die Chatsitzung schließt. Die Chatfenster sollten daraufhin geschlossen werden.

Siehe auch

[Übersicht Events](#) | [EChatEnd](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.7 OnClientClose

Der Client soll geschlossen und auf den Neustart vorbereitet werden.

```
Event OnClientClose(  
    Long IMinutes  
)
```

Parameter

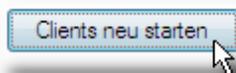
IMinutes

Zeitspanne in Minuten, nach deren Verlauf der Client neu gestartet werden soll, wenn der Wert positiv oder 0.

Bemerkungen

Diese Nachricht schickt der Administrator, wenn er wichtige Dateien austauschen will. Das macht Sinn wenn die Clients aus einem freigegebenen Netzwerkverzeichnis gestartet wurden. Dazu müssen alle Clients heruntergefahren werden. Wenn Sie dieses Feature unterstützen möchten rufen Sie hier [NPrepareRestart](#) auf und übergeben *IMinutes* als Parameter. Bei einem negativen Wert von *IMinutes* soll die Anwendung nicht neu gestartet werden. Schließen Sie danach Ihre Anwendung.

OnClientClose wird über die Serverkonsole ausgelöst (Seite Benutzerverwaltung).

**Siehe auch**

[Übersicht Events](#) | [NPrepareRestart](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.8 OnConfNotify

Die Konferenzzugehörigkeit hat sich geändert.

```
Event OnConfNotify(  
    Bool bConf  
)
```

Parameter

bConf

True, wenn der Benutzer in eine Konferenz eintritt, False, wenn er sie verlässt.

Bemerkungen

Beim Eintreten in eine und beim Verlassen von Konferenzen wird diese Nachricht gesendet.

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.9 OnConnectionState

Der Zustand der Verbindung hat sich geändert.

```
Event OnConnectionState(  
    Long IState  
)
```

Parameter

IState

Zustand, in dem sich die Verbindung zum Server befindet.

Bemerkungen

Der Zustand der Netzwerkverbindung hat sich geändert.

Mögliche Werte für den Zustand sind:

0: Unbekannt

1: Keine Netzwerkverbindung

2: Verbindung steht, aber Server arbeitet nicht.

3: Verbindung steht, CTI-Server arbeitet.

Der Verbindungszustand ändert sich beim Verbinden des Clients mit dem Server oder wenn der Server heruntergefahren wird bzw. der Service eine Pause einlegt.

Beispiel

[Mini CTI Client \(Delphi\)](#)

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.10 OnDatabaseToUser

Eine benutzerdefinierte Nachricht wurde vom Server geschickt.

```
Event OnDatabaseToUser(  
    Long ICommand,  
    String bstrMessage  
)
```

Parameter*ICommand*

Typ des benutzerdefinierten Kommandos

bstrMessage

Übermittelte Parameter

Bemerkungen

TAPIMaster® besitzt auf der Serverseite eine zweite Schnittstelle. Normalerweise werden Datenbankabfragen über die ODBC-Schnittstelle des Servers getätigt. Alternativ dazu kann die Serverschnittstelle jedoch in einer Datenbank oder anderen Anwendung eingebunden werden. Über [EUserSendToDataBase](#) können dann benutzerdefinierte Daten getauscht werden.

Siehe auch[Übersicht Events](#) | [EUserSendToDataBase](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.11 OnDbBookRecord

Ein Datensatz aus der Rufnummernsuche wird übermittelt.

```
Event OnDbBookRecord(  
    String bstrName,  
    String bstrNumber,  
    String bstrExtra  
)
```

Parameter*bstrName*

Name des Teilnehmers

bstrNumber

Seine Rufnummer

bstrExtra

Zusätzliche Informationen

Bemerkungen

Der Benutzer hat eine Anfrage an die Datenbank vom Typ [SQLGetDbBook](#) gestartet. Die Datensätze kommen Record für Record. bstrExtra kann leer sein. Umfang und Format der ankommenden Daten werden in den Datenbankeinstellungen des CTI-Servers festgelegt.

Siehe auch[Übersicht Events](#) | [SQLGetDbBook](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.12 OnForwardingState

Die eigene Rufumleitung wird übertragen.

```
Event OnForwardingState(  
    Bool bSet,  
    String bstrNumber  
)
```

Parameter

bSet

Wenn True, dann ist eine Rufumleitung geschaltet.

bstrNumber

Nummer, zu der umgeleitet wird.

Bemerkungen

Die Einstellungen für die Rufumleitungen werden zentral auf dem CTI-Server gespeichert. Deshalb wird diese Nachricht nach der Anmeldung gesendet. Bei Änderungen an der Rufumleitung wird **OnForwardingState** ebenfalls aufgerufen.

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Events](#) | [TSetForwarding](#) | [TForwardingState](#) | [TForwardingDest](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.13 OnGroupMember

Teilt den Clients beim Start die Mitglieder ihrer Gruppe und deren Zustand mit.

```
Event OnGroupMember(  
    String bstrNumber,  
    String bstrName,  
    Bool bOnline,  
    Bool bActiveCall,  
    Bool bOfferings  
    Bool bRedirection  
)
```

Parameter

bstrNumber

Durchwahl des Gruppenmitglieds, 2 bis 4 Stellen.

bstrName

Name des Gruppenmitglieds. Hier wird der Anmeldename verwendet. Dieser ist erst verfügbar wenn sich dieser Kollege mindestens einmal eingeloggt hat, da sich diese Daten automatisch einpflegen.

bOnline

Ist True, wenn der Teilnehmer gerade eingeloggt ist.

bActiveCall

Ist True, wenn der Teilnehmer gerade aktive Gespräche hat.

bOfferings

Ist True, wenn unbeantwortete Anrufe beim Teilnehmer in der Warteschlange stehen. Diese Anrufe können bei vielen Anlagen herangeholt werden.

bRedirection

Ist True, wenn der Teilnehmer eine Rufumleitung geschaltet hat.

Bemerkungen

Bei jedem Anmelden werden die Daten der Gruppenmitglieder geschickt. Pro Gruppenmitglied wird eine Nachricht gesendet. Wird das Event mit einem leeren String in *bstrName* aufgerufen, so ist die Übertragung beendet. Mit diesen Events sind gleich zu Programmstart die Gesprächszustände der Gruppenmitglieder bekannt. Diese werden zur Laufzeit mit [OnSendUserState](#) aktualisiert.

Siehe auch

[Übersicht Events](#) | [OnSendUserState](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.14 OnIdentity

Die Rufnummernidentifikation kommt an.

```
Event OnIdentity(  
    Long ICallID,  
    String bstrName  
)
```

Parameter

ICallID

Anruf-ID des Gesprächs

bstrName

Name des Anrufers, bis zu 255 Zeichen

Bemerkungen

Auf dem Server wurde eine Rufnummernidentifikation durchgeführt. Diese kann von der ODBC-Schnittstelle oder der Server Programmierschnittstelle kommen. Neben dem Namen können in *bstrName* noch weitere Daten übertragen werden.

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Events](#) | [SendIdentity](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.15 OnInvalidStringFormat

Server und Client passen von der Plattform her nicht zueinander.

Event OnInvalidStringFormat()**Parameter**

Keine

Bemerkungen

Es gibt eine TAPIMaster® Version für ANSI-Standard und eine für Unicode. Letztere arbeitet intern mit 2Byte Zeichen. Client und Server müssen immer der gleichen Version angehören. In Deutschland wurde bisher nur die ANSI-Version vertrieben, so dass der Fehler nicht auftreten sollte.

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.16 OnLicenseInfo

Liefert erweiterte Lizenzinformationen.

**Event OnLicenseInfo(
String bstrData
)****Parameter**

bstrData

Erweiterte Lizenzinformationen. Die übermittelten Werte sind von geschweiften Klammern eingerahmt und durch Semikolons getrennt. Formatbeispiel:
{License for: Musterfirma;Street: Musterstraße 1;City: 12345 Königs-Musterhausen;Phone: +49 (5555) 664488;Web: http://www.musterfirma.de}

Bemerkungen

Die Nachricht wird nach dem Anmelden gesendet.

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.17 OnLocaleInfo

Liefert Anlagenparameter und Landeseinstellungen.

**Event OnLocaleInfo(
Long IInternLength,
Long ICountry,
String bstrCity,
String bstrPBX**

)

Parameter*lInternLength*

Maximale Länge interner Durchwahlen, Werte von 2 bis 4 sind möglich.

lCountry

Landeskennung, z. B. 49 für Deutschland

bstrCity

Vorwahl ohne führende 0

bstrPBX

Anlagenrufnummer. Wird bei Telefonanlagen mit Amtsanschluss verwendet. Bei einer Rufnummer +49 (69) 123456-789 wäre 123456 die Anlagenrufnummer.

Bemerkungen

Die Funktion liefert bei jeder Anmeldung diese lokalen Parameter zurück, die dann im Programm weiter verwendet werden können.

Siehe auch[Übersicht Events](#)[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.18 OnLogin

Die Anmeldung am CTI-Server wird bestätigt.

```

Event OnLogin(
  Bool bSuccess
)

```

Parameter*bSuccess*

Erfolgsstatus der Anmeldung, ist True wenn der Benutzer sich anmelden konnte.

Bemerkungen

Eine erfolgreiche Anmeldung ist die Voraussetzung dafür dass die meisten Befehle abgesetzt werden können und Events empfangen werden.

Siehe auch[Übersicht Events](#) | [NLogin](#)[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.19 OnMenuCommand

Das Menü des Verbindungssymbols wurde angeklickt.

```

Event OnMenuCommand(

```

UShort wCommand

)

Parameter

wCommand
Befehls-ID

Bemerkungen

Der Benutzer hat auf das Menü des Verbindungssymbols geklickt. Der ausgelöste Befehl wird in *wCommand* übertragen.

Siehe auch

[Übersicht Events](#) | [OConnectionIconOptions](#) | [OShowIcon](#) | [OShowMenu](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.20 OnObjectOfRedirect

Auf den Benutzer wird eine Rufumleitung geschaltet.

Event OnObjectOfRedirect(

Bool bSet,
String bstrNumber

)

Parameter

bSet

Auf den Teilnehmer wurde eine Rufumleitung geschaltet, wenn True. Wenn False wurde die Rufumleitung wieder abgeschaltet.

bstrNumber

Durchwahl des Teilnehmers, der zu diesem Anschluß umleitet.

Bemerkungen

Wenn ein anderer Teilnehmer eine Rufumleitung hierher schaltet wird dieser Event aufgerufen. Ebenso wenn dieser die Rufumleitung wieder herausnimmt. Sollten Rufumleitungen auf diesen Anschluß geschaltet sein so wird die Meldung auch nach Anmeldung übertragen.

Siehe auch

[Übersicht Events](#) | [TSetForwarding](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.21 OnProtocolData

Gesprächsdatensätze werden übermittelt.

Event OnProtocolData(

String bstrOther,
String bstrOwn,

**Long IStart,
Long IEnd,
Bool bSuccess,
Bool blnbound**

)

Parameter

bstrOther

Rufnummer des Gesprächspartners

bstrOwn

Eigene Durchwahl

IStart

Gesprächsbeginn, Format time_t

IEnd

Gesprächsende, Format time_t

bSuccess

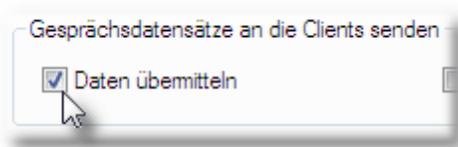
Ist True, wenn das Gespräch erfolgreich war, beide Partner waren miteinander verbunden.

blnbound

Ist True, wenn das Gespräch eingehend war, bei ausgehenden Gesprächen ist *blnbound* False.

Bemerkungen

Nach Ende jedes Gesprächs wird ein solcher Gesprächsdatensatz übertragen. Ist der Benutzer abgemeldet, so werden die Daten auf dem CTI-Server gepuffert und bei der nächsten Anmeldung gesendet. Die Daten werden nur gesendet, wenn auf dem Server diese Option eingeschaltet ist. Auf der Seite "Datenbankschnittstelle" der Serverkonsole kann die Übermittlung zugeschaltet werden.



Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.22 OnRaiseClient

Teilt dem Client mit dass er sich in den Vordergrund stellen soll.

Event OnRaiseClient()

Parameter

Keine

Bemerkungen

Ein Doppelklick auf das Netzwerksymbol löst dieses Event aus. Der Benutzer möchte die Telefonieanwendung im Vordergrund sehen.

Siehe auch

[Übersicht Events](#) | [OShowIcon](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.23 OnSendAddress

Die Adresse, unter welcher der Anruf einging wird gesendet.

```
Event OnSendAddress(  
    Bool blsAddress,  
    Long lCallID,  
    String bstrAddress  
)
```

Parameter

blsAddress

Wenn True dann wird die Adresse des Telefons in *bstrAddress* übermittelt, bei False ist es die Rufnummer.

lCallID

Anruf-ID des Gesprächs

bstrAddress

Adresse welche angerufen wurde oder Rufnummer.

Bemerkungen

Diese Nachricht wird bei eingehenden Anrufen gesendet. Ein ISDN-Telefon kann auf mehrere MSNs reagieren. Nehmen wir z. B. einen Heimarbeitsplatz. Hier wird es eine MSN für Privatgespräche und eine MSN für dienstliche Anrufe geben. Am Monatsende soll aber ausgewertet werden, wieviel Zeit dienstlich telefoniert wurde. Über die angerufene MSN in *bstrAddress* können die Anrufe zugeordnet werden.

Eine andere Möglichkeit trifft man bei Telefonanlagen. Eine Leitung kann mehrere interne Durchwahlen haben. Die 210 kann für normale und interne Gespräche verwendet werden, die 310 ist mit einer Hotline verbunden. In *bstrAddress* würde entweder die 210 oder die 310 mitgeliefert.

TAPIMaster® versucht, eine der beiden Angaben zu ermitteln und zu senden. Adressen werden jedoch von nur von wenigen Treibern unterstützt. Was gesendet wird, ist von der Anlage bzw. dem Telefon und den Treibern abhängig. Es ist auch möglich, dass diese Nachricht nicht gesendet wird oder mehrmals während eines Anrufs kommt.

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.24 OnSendUserState

Der Status eines Gruppenmitglieds hat sich geändert.

```
Event OnSendUserState(  
    String bstrNumber,  
    Bool bLogin,  
    Bool bActiveCall,  
    Bool bOfferings,  
    Bool bRedirection  
)
```

Parameter

bstrNumber

Durchwahl des Gruppenmitglieds

bLogin

Ist True wenn das Gruppenmitglied angemeldet ist.

bActiveCall

Ist True, wenn das Gruppenmitglied gerade ein Gespräch führt.

bOfferings

Ist True wenn das Gruppenmitglied angerufen wird und diesen Anruf noch nicht angenommen hat.

bRedirection

Ist True, wenn das Gruppenmitglied Anrufe umleitet.

Bemerkungen

Die Nachricht wird übertragen, wenn sich der Zustand eines Gruppenmitglieds geändert hat. Der Benutzer hat somit die Möglichkeit zu reagieren wenn ein Gruppenmitglied ein Gespräch nicht annimmt oder ein anklopfendes Gespräch erhält. Seine Erreichbarkeit kann ebenfalls immer eingesehen werden.

Beispiel

[Einbindung in Excel und VB](#)

Siehe auch

[Übersicht Events](#) | [OnGroupMember](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.25 OnServerMessage

Der Administrator schickt dem Benutzer eine Nachricht.

```
Event OnServerMessage(  
    String bstrMessage  
)
```

Parameter

bstrMessage
Nachrichtentext

Bemerkungen

Der Administrator schickt eine Nachricht an den Benutzer, welche in *bstrMessage* enthalten ist. Es werden bis max. 255 Zeichen gesendet. Die Nachrichten werden von der Serverkonsole auf der Seite "Benutzerverwaltung" gesendet.

**Siehe auch**

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.26 OnShow PhoneList

Der Benutzer hat einen Doppelklick auf das gelbe Telefonsymbol getätigt.

Event OnShowPhoneList()**Parameter**

Keine

Bemerkungen

Bei eingehenden Anrufen kann ein gelbes Telefonsymbol in der Taskleiste angezeigt werden. Ein Doppelklick darauf löst diesen Event aus. Das Programm sollte dann daraufhin die Anruferliste anzeigen, falls es eine solche Liste gibt.

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.27 OnSQLData

SQL-Abfragen werden beantwortet.

```
Event OnSQLData(  
    ULong dwRequest,  
    Long ICount,  
    String bstrData  
)
```

Parameter

dwRequest

Nummer der Anfrage, welche mit [SQLBegin](#) gestartet wurde.

ICount

Anzahl der zurückgelieferten Datensätze, ist 0 wenn keine Daten vorhanden waren und -1 wenn die Abfrage fehlschlug.

bstrData

Text mit den angeforderten Daten.

Bemerkungen

Dieser Event wird immer dann aufgerufen wenn Daten mit [SQLBegin](#) oder [SQLNext](#) angefordert wurden. Die Daten können im XML-Format oder als Text kommen. Die Abfrage kann fehlschlagen wenn eine Abfrage bereits terminiert wurde oder bereits alle Daten gelesen wurden.

Beispiel

[Datenbankabfragen](#)

Siehe auch

[Übersicht Events](#) | [SQLBegin](#) | [SQLNext](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.28 OnSQLEnd

Eine SQL-Abfrage wurde beendet.

```
Event OnSQLEnd(  
    ULong dwRequest,  
    Bool bSuccess  
)
```

Parameter*dwRequest*

Nummer der Datenbankabfrage, welche bei [SQLBegin](#) vergeben wurde.

bSuccess

Ist True wenn die Abfrage beendet werden konnte.

Bemerkungen

Der Event wird auf den Befehl [SQLEnd](#) hin gesendet. Werden mit einem [SQLBegin](#) oder [SQLNext](#) alle Datensätze gelesen, so kommt dieser Event selbständig. Wird eine Datenbankabfrage nicht terminiert, so terminiert der Server diese automatisch nach einigen Minuten und sendet diesen Event.

Siehe auch

[Übersicht Events](#) | [SQLEnd](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.29 OnSQLLogin

Die Anmeldung an die Datenbank wird bestätigt.

```
Event OnSQLLogin(  
    Bool bSuccess  
)
```

Parameter*bSuccess*

Ist True, wenn die Anmeldung erfolgreich war.

Bemerkungen

Bevor ein Anwender die Datenbank benutzt muss er sich dort mit [SQLLogin](#) anmelden. Der Server schickt daraufhin diese Nachricht zurück. Wenn bei der Anmeldung ein falsches Paßwort übermittelt wurde oder die Datenbankverbindung nicht aktiv ist wird die Anmeldung abgewiesen. Die Datenbankverbindung wird auf der Serverkonsole auf der Seite "Datenbankschnittstelle" zugeschalten.

**Siehe auch**

[Übersicht Events](#) | [SQLLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.2.6.30 OnUserToUser

Ein anderer Benutzer schickt eine Nachricht. Nur in der Netzwerkversion vorhanden.

```
Event OnUserToUser(  
    String bstrOther,  
    String bstrMessage  
)
```

Parameter*bstrOther*

Durchwahl des anderen Teilnehmers

bstrMessage

Textnachricht, die er verschickt, maximal 255 Zeichen.

Bemerkungen

Die Benutzer können sich untereinander mit [ESendUserToUser](#) Nachrichten zukommen lassen.

Wenn Sie diese Funktion implementieren können Sie möglicherweise an manchen Arbeitsplätzen auf einen Mailclient verzichten.

Siehe auch

[Übersicht Events](#) | [ESendUserToUser](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3 C++ CTI API

Die Schnittstelle für die C++ API befindet sich in der Datei FCTICLNT.DLL. Sie benötigen weiterhin die Dateien CTICLNT.H und CTYPES.H sowie FCTICLNT.LIB aus dem CTI SDK. Kopieren Sie diese Dateien in die zugehörigen Verzeichnisse Ihrer Entwicklungsumgebung. Führen Sie auf dem Zielrechner ein Clientsetup aus, damit dort alle notwendigen Dateien vorhanden bzw. registriert sind.

[Netzwerk](#)

[Optionen](#)

[Telefoniefunktionen](#)

[SQL](#)

[Zusatzfunktionen](#)

[Events](#)

[Strukturen](#)

[Konstanten](#)

Siehe auch

[Befehlsreferenz Client](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1 Netzwerk

Verwendung der Netzwerkfunktionen

Die Netzwerkfunktionen dienen der Steuerung der Verbindung zum CTI-Server.

Präfix

Die Netzwerkfunktionen haben den Präfix "CTI_N" vorangestellt.

Serveridentifikation

CTI_NGetServerName	Namen des Servers abfragen.
CTI_NSetServerName	Namen des Servers ändern.
CTI_NGetPortNumber	Portnummer abfragen.
CTI_NSetPortNumber	Portnummer ändern.
CTI_NSetNetworkOptions	Netzwerkverbindungsoptionen ändern, keine neue Verbindung.
CTI_NNetworkConfigDialog	Konfigurationsdialog für Netzwerkverbindungen anzeigen.

Steuerung der Netzwerkverbindung

CTI_N_StartClientCALLBACK	Initialisierung der Schnittstelle, Output über Callback Funktionen.
CTI_NStartClientHWND	Initialisierung der Schnittstelle, Output über Windows-Nachrichten.
CTI_NLogin	Mit Benutzernamen am CTI Server anmelden.
CTI_NLogoff	Vom CTI Server abmelden.
CTI_NClientRestart	Netzwerkverbindung neu herstellen.
CTI_NClientStop	Netzwerkverbindung beenden.
CTI_NClientActive	Abfragen ob Netzwerkverbindung aktiv ist.
CTI_NPrepareRestart	Neustart den Clients vorbereiten.
CTI_NIsNetworkVersion	Ist die verwendete DLL die Netzwerkversion?

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.1 CTI_NClientActive

Zeigt den Status der Netzwerkverbindung an.

```
BOOL WINAPI CTI_NClientActive();
```

Parameter

Keine

Rückgabewert

BOOL

TRUE oder FALSE, je nachdem ob die Netzwerkverbindung steht oder nicht.

Bemerkungen

Die Netzwerkverbindung wird mit [CTI_N_StartClientCALLBACK](#) oder [CTI_NStartClientHWND](#) initialisiert. Sie kann mit [CTI_NClientStop](#) unterbrochen und mit [CTI_NClientRestart](#) wieder aufgenommen werden.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [WM_CTI_NETWORKCLOSE](#) | [WM_CTI_NETWORKOPEN](#) | [CTI_SERVICEPAUSE](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.2 CTI_NClientRestart

Die Netzwerkverbindung wird wieder hergestellt.

```
VOID WINAPI CTI_NClientRestart();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Stellt eine Netzwerkverbindung wieder her die mit [CTI_NClientStop](#) unterbrochen wurde.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NClientStop](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.3 CTI_NClientStop

Die Netzwerkverbindung wird unterbrochen.

```
VOID WINAPI CTI_NClientStop();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Mit dieser Funktion können Sie die Netzwerkverbindung zum CTI-Server beenden. Wird automatisch beim Entladen der Clientschnittstelle aufgerufen.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NClientRestart](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.4 CTI_NGetPortNumber

Liefert die Portnummer der Netzwerkverbindung zurück.

```
LONG WINAPI CTI_NGetPortNumber();
```

Parameter

Keine

Rückgabewert

LONG

Portnummer der Netzwerkverbindung

Bemerkungen

Die Portnummer lässt sich mit [CTI_NSetPortNumber](#) ändern.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NNetworkConfigDialog](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.5 CTI_NGetServerName

Liefert den Servernamen der Netzwerkverbindung zurück.

```
PTCHAR WINAPI CTI_NGetServerName();
```

Parameter

Keine

Rückgabewert

PTCHAR

Nullterminierte Zeichenkette mit dem Namen des Servers

Bemerkungen

Der Servername lässt sich mit **CTI_NGetServerName** ändern. Statt des Servernamens kann auch die IP-Adresse oder "localhost" zurückgeliefert werden.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.6 CTI_NIsNetworkVersion

Info über den Typ der FCTICLNT.DLL.

```
BOOL WINAPI CTI_NIsNetworkVersion();
```

Parameter

Keine

Rückgabewert

BOOL

TRUE oder FALSE, je nachdem ob die verwendete FCTICLNT.DLL in der Netzwerkversion vorliegt oder nicht.

Bemerkungen

Es gibt nur noch eine Netzwerkversion, die Funktion existiert nur aus Kompatibilitätsgründen.

Siehe auch

[Übersicht Netzwerkfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.7 CTI_NLogin

Meldet Client am CTI-Server an.

```
VOID WINAPI CTI_NLogin();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Die Funktion wird automatisch ausgeführt, wenn [CTICLIENT_AUTOLOGIN](#) aktiv ist und die Netzwerkverbindung steht. **CTI_NLogin** schaltet im CTI-Server die Funktionalität für den Client frei. Die anderen Gruppenmitglieder sehen den Client dann als angemeldet. Wenn Sie die Schnittstelle in ein Programm einbinden was den ganzen Tag über läuft kann es sinnvoll sein die CTI-Funktionalität

zeitweilig abzuschalten, z. B. während der Mittagspause. Die anderen Gruppenmitglieder sehen dann dass der Benutzer nicht ansprechbar ist. [CTI_NLogoff](#) schaltet die CTI ab und **CTI_NLogin** wieder an.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTICLIENT_AUTOLOGIN](#) | [CTI_NLogoff](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.8 CTI_NLogoff

Meldet Client vom CTI-Server ab.

```
VOID WINAPI CTI_NLogoff();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Schaltet die CTI-Funktionalität im Server ab. Die Gruppenmitglieder sehen den Client dann als abgemeldet. Alternativ dazu kann auch die Netzwerkverbindung komplett mit [CTI_NClientStop](#) gekappt werden.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.9 CTI_NNetworkConfigDialog

Zeigt Netzwerkdialog an.

```
VOID WINAPI CTI_NNetworkConfigDialog();
```

Parameter

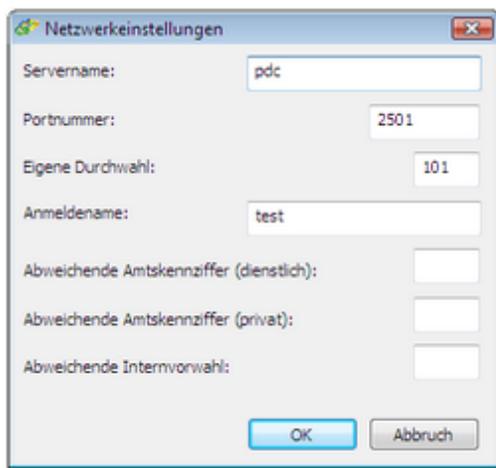
Keine

Rückgabewert

Keiner

Bemerkungen

Ein Dialog wird geöffnet in dem die Verbindungsdaten eingestellt werden können. Alternativ dazu ist die Verwendung von [CTI_NSetNetworkOptions](#) und [CTI_OSetOwnerNumber](#) möglich. Fehlen die ersten drei Angaben und wird die Verbindungsoption CTICLIENT_CHECKCONNECTION verwendet, so erscheint der Dialog automatisch beim Programmstart. Für die Anzeige des Dialogs wird die Verbindung zum CTI-Server unterbrochen. Mit OK wird eine neue Verbindung hergestellt. Benutzen Sie diesen Dialog wenn Sie nach einer Möglichkeit suchen die Verbindungsparameter zu ändern und selbst keine Eingabemöglichkeit programmieren wollen.



Beispiel

[SQL-Demo](#)

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_OSetOptions](#) | [CTI_NSetNetworkOptions](#),

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.10 CTI_NPrepareRestart

Bereite Client auf Neustart vor.

```
VOID WINAPI CTI_NPrepareRestart(
    LONG IMinutes
);
```

Parameter

IMinutes

Zeit in Minuten, die bis zum Neustart vergehen soll. Bei *IMinutes* = 0 erfolgt der Neustart sofort, bei negativen Werten wird der Client nicht neu gestartet.

Rückgabewert

Keiner

Bemerkungen

Der Server kann die Standardclients herunterfahren und wieder neu starten. Das kann sinnvoll sein wenn die Clients aus einem Netzwerk heraus gestartet wurden und nun die Dateien ausgetauscht werden sollen. Über die Serverkonsole auf der Seite "Benutzerverwaltung" kann der Neustart angestoßen werden.



Sie können diese Funktionen ebenfalls unterstützen. Wenn ein Event [CTI_CLIENTCLOSE](#) eintritt, so bekommen Sie in dessen IParam die Zeit mitgeteilt, in welcher der Client neu starten soll. Rufen Sie jetzt [CTI_NPrepareRestart](#) mit dieser Zeit als Parameter auf und beenden Sie die Anwendung. Sie wird nach der verstrichenen Zeit neu gestartet.

Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_CLIENTCLOSE](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.11 CTI_NSetNetworkOptions

Startet Netzwerkverbindung mit geänderten Verbindungsparametern neu.

```
VOID WINAPI CTI_NSetNetworkOptions(  
    PDWORD pdwPort,  
    PTCHAR pszServerName  
);
```

Parameter

pdwPort

Port auf dem CTI-Server, zu dem die Verbindung aufgebaut werden soll.

pszServerName

Name des CTI-Servers oder dessen IP-Adresse.

Rückgabewert

Keiner

Bemerkungen

Die Funktion ändert die Verbindungsparameter. Die bestehende Netzwerkverbindung zum CTI-Server wird unterbrochen und mit den neuen Verbindungsdaten wieder aufgebaut.

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NSetPortNumber](#) | [CTI_NSetServerName](#) | [CTI_NClientRestart](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.12 CTI_NSetPortNumber

Die Portnummer der Netzwerkverbindung wird geändert.

```
VOID WINAPI CTI_NSetPortNumber(  
    LONG IPort  
);
```

Parameter

IPort

Die Portnummer der Verbindung.

Rückgabewert

Keiner

Bemerkungen

Die Funktion ändert nur die Portnummer, baut jedoch keine neue Verbindung auf. Um eine neue Verbindung mit geänderten Parametern aufzubauen verwenden Sie [CTI_NClientRestart](#).

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NSetNetworkOptions](#) | [CTI_NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.13 CTI_NSetServerName

Der Servername der Netzwerkverbindung wird geändert.

```
VOID WINAPI CTI_NSetServerName(  
    PTCHAR pszServerName  
);
```

Parameter

pszServerName

Der Name des Servers, seiner IP-Adresse oder "localhost" werden übergeben.

Rückgabewert

Keiner

Bemerkungen

Die Funktion ändert nur den Namen, baut jedoch keine neue Verbindung auf. Um eine neue Verbindung mit geänderten Parametern aufzubauen verwenden Sie [CTI_NClientRestart](#).

Siehe auch

[Übersicht Netzwerkfunktionen](#) | [CTI_NSetNetworkOptions](#) | [CTI_NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.14 CTI_NStartClientCALLBACK

Die Schnittstelle wird initialisiert.

```
VOID WINAPI CTI_NStartClientCALLBACK(  
    InfoParentProc pProc  
);
```

Parameter

pProc

Adresse einer Callbackfunktion, welche folgendes Format hat:

```
typedef VOID WINAPI CallbackFunc(DWORD, WPARAM, LPARAM);
```

Handelt es sich bei der Callbackfunktion um eine Objektfunktion, so muss diese statisch sein.

Rückgabewert

Keiner

Bemerkungen

Mit dieser Funktion wird die Schnittstelle initialisiert. Alle Events der Schnittstelle laufen dann auf der Callbackfunktion auf. Die Funktion darf nicht zusammen mit [CTI_NStartClientHWND](#) verwendet werden.

Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Netzwerkfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.1.15 CTI_NStartClientHWND

Die Schnittstelle wird initialisiert.

```
VOID WINAPI CTI_NStartClientHWND(  
    HWND hWndParent  
);
```

Parameter

hWndParent

Fensterhandle, welches die Events empfängt.

Rückgabewert

Keiner

Bemerkungen

Mit dieser Funktion wird die Schnittstelle initialisiert. Alle Events der Schnittstelle laufen dann auf der zugehörigen Fensterfunktion auf. Die Funktion darf nicht zusammen mit [CTI_NStartClientCALLBACK](#) verwendet werden.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht Netzwerkfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2 Optionen

Verwendung der Optionen

Die Optionen bestimmen das Verhalten der CTI Schnittstelle. Sie werden geschaltet bevor die Schnittstelle mit [CTI_NStartClientCALLBACK](#) oder [CTI_NStartClientHWND](#) initialisiert wird. Die Programmoptionen werden von der Schnittstelle intern gespeichert, es ist nicht erforderlich dass Sie diese Werte selbst sichern. Die Startwerte holt sich die Schnittstelle bei der [Installation](#) aus der Datei FCTIINST.INI, falls diese nicht vorhanden ist werden Voreinstellungen verwendet. Sie benötigen die Optionsfunktionen normalerweise nicht, es sei denn Ihr Programm ermöglicht es diese zu ändern. Nutzen Sie die Schnittstelle für verschiedene CTI-Programme auf dem gleichen Rechner, so sollte jedes Programm seine Standardeinstellungen explizit setzen. Die meisten Optionen außer der internen Durchwahl können auch mit dem [Setup-Konfigurator](#) festgelegt werden. Er schreibt die FCTIINST.INI.

Präfix

Die Optionsfunktionen haben den Präfix "CTI_O" vorangestellt.

Setzen der Bitflags

CTI_OGetOptions	Liefert die Optionen als Bitflag zurück.
CTI_OSetOptions	Legt die Parameter als Bitflag fest.

Parameteroptionen

CTI_OConnectionIconOptions	Das Aussehen des Verbindungssicons wird geändert.
CTI_OGetUserCallInfo	Zeigt, wie der Client über einen neuen Anruf informiert wird.
CTI_OSetUserCallInfo	Legt fest, wie der Benutzer bei eingehenden Anrufen informiert wird.
CTI_OShellIconOptions	Veraltet. Verwenden Sie CTI_OConnectionIconOptions .
CTI_OGetDialKey	Liefert die Tastenkombination zum Anwählen von markiertem Text zurück.
CTI_OSetDialKey	Legt die Tastenkombination zum Anwählen von markiertem Text fest.
CTI_OGetHotKey	Veraltet. Verwenden Sie CTI_OGetDialKey .
CTI_OSetHotKey	Veraltet. Verwenden Sie CTI_OSetDialKey .
CTI_OGetDropKey	Liefert die Tastenkombination zum Auflegen eines Gesprächs zurück.
CTI_OSetDropKey	Legt die Tastenkombination zum Auflegen von Gesprächen fest.
CTI_OGetIniFileName	Liefert den Namen der Konfigurationsdatei zurück.
CTI_O_GetOwnerNumber	Liefert die eigene Durchwahl zurück. Nur in der Netzwerkversion vorhanden.
CTI_OSetOwnerNumber	Legt die eigene Durchwahl fest. Nur in der Netzwerkversion vorhanden.

CTI_OGetOwnerName	Liefert den eigenen Anmeldenamen zurück.
CTI_OSetOwnerName	Legt den Anmeldenamen fest.

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.1 CTI_OConnectionIconOptions

Das Aussehen des Verbindungsicons wird geändert.

```
VOID WINAPI CTI_OConnectionIconOptions(
    HICON hIconOn,
    HICON hIconOff,
    HMENU hMenu,
    PTCHAR pszText
);
```

Parameter

hIconOn

Handle des Icons, welches bei bestehender Netzwerkverbindung zum CTI Server angezeigt wird. Bei 0 wird das Standardicon angezeigt.

hIconOff

Handle des Icons, welches bei unterbrochener Netzwerkverbindung zum CTI Server angezeigt wird. Bei 0 wird das Standardicon angezeigt.

hMenu

Handle auf ein Menü, welches über dem Verbindungsicon angezeigt wird, wenn dieser mit der rechten Maustaste darauf klickt. Kann 0 sein, dann wird kein Menü verwendet.

pszText

Tooltip, welcher über dem Verbindungsicon angezeigt wird. Maximal 63 Zeichen. Kann Null sein, dann wird ein Standardtext angezeigt. Soll kein Tooltip angezeigt werden, so übergeben Sie einen Leerstring.

Rückgabewert

Keiner

Bemerkungen

Aussehen, Menü und Tooltips des Verbindungsicons lassen sich ändern. Um das Verbindungsicon nutzen zu können, muss die Option [CTICLIENT_SHOWICON](#) aktiviert sein. Bei Menüs ist zusätzlich [CTICLIENT_SHOWMENU](#) erforderlich.

Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Optionen](#) | [CTI_OSetOptions](#) | [CTICLIENT_SHOWICON](#) | [CTICLIENT_SHOWMENU](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.2 CTI_OGetDialKey

Liefert die Tastenkombination zum Anwählen von markiertem Text zurück.

LONG WINAPI CTI_OGetDialKey();

Parameter

Keine

Rückgabewert

LONG

Im niederwertigen Word ist die Hotkey-Taste, im höherwertigen Word die Kombination aus den zusätzlichen Steuertasten enthalten. Diese Steuertasten können sein:

0x00 = Keine weitere Taste

0x01 = Umschalttaste

0x02 = Steuertaste

0x03 = Umschalt- und Steuertaste

Bemerkungen

Die Werte für den Hotkey entsprechen dem virtuellen Keycode der F-Taste, dieser kann von 0x70 (F1) bis 0x7B (F12) liegen. Der Code zum Wählen sollte nicht mit dem Code zum Auflegen identisch ein.

Siehe auch

[Übersicht Optionen](#) | [CTICLIENT_HOTKEY](#) | [CTI_OSetDialKey](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.3 CTI_OGetDropKey

Liefert die Tastenkombination zum Auflegen eines Gesprächs zurück.

LONG WINAPI CTI_OGetDropKey();

Parameter

Keine

Rückgabewert

LONG

Im niederwertigen Word ist die Taste zum Auflegen, im höherwertigen Word die Kombination aus den zusätzlichen Steuertasten enthalten. Diese Steuertasten können sein:

0x00 = Keine weitere Taste

0x01 = Umschalttaste

0x02 = Steuertaste

0x03 = Umschalt- und Steuertaste

Bemerkungen

Die Werte für den Dropkey entsprechen dem virtuellen Keycode der F-Taste, dieser kann von 0x70 (F1) bis 0x7B (F12) liegen. Der Code zum Wählen sollte nicht mit dem Code zum Auflegen identisch ein.

Siehe auch

[Übersicht Optionen](#) | [CTICLIENT_HOTKEY](#) | [CTI_OSetDropKey](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.4 CTI_OGetIniFileName

Liefert den Namen der Konfigurationsdatei zurück.

```
PTCHAR WINAPI CTI_OGetIniFileName();
```

Parameter

Keine

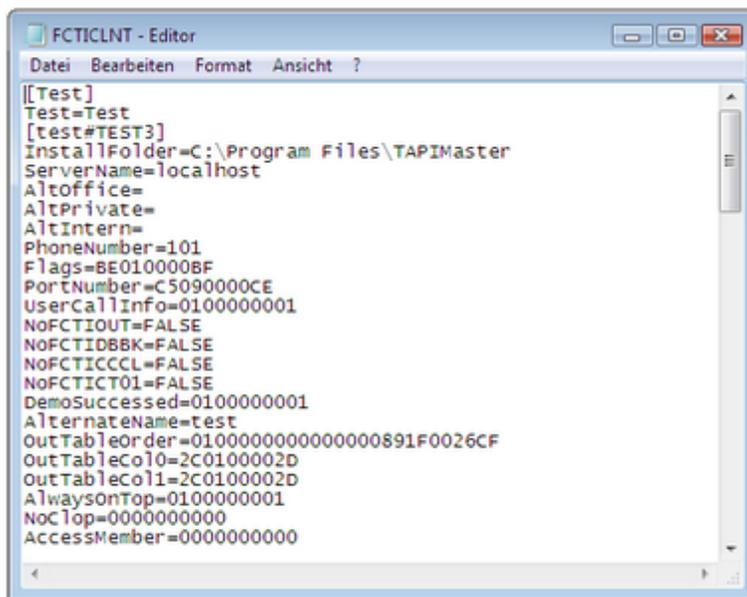
Rückgabewert

PTCHAR

Nullterminierter String mit dem Namen der Konfigurationsdatei der CTI-Clientschnittstelle.

Bemerkungen

Die CTI-Schnittstelle speichert ihre Einstellungen in einer Konfigurationsdatei. Wenn Sie als Programmierer für die Einstellungen Ihres Clients keine separate Datei anlegen wollen, können Sie diese verwenden. Die Datei liegt im Windows-INI Format vor. Der Sektionsname ergibt sich aus dem Benutzernamen + # + Computernamen. Beispiel: [mueller#wk12]. Werfen Sie einen Kurzen Blick in eine der vorhandenen INI-Dateien, um keine vorhandenen Schlüsselwörter zu benutzen.



```
FCTICLNT - Editor
Datei Bearbeiten Format Ansicht ?
[Test]
Test=Test
[test#TEST3]
InstallFolder=C:\Program Files\TAPIMaster
ServerName=localhost
AltOffice=
AltPrivate=
AltIntern=
PhoneNumber=101
Flags=BE010000BF
PortNumber=C5090000CE
UserCallInfo=0100000001
NoFCTIOUT=FALSE
NoFCTIDBBK=FALSE
NoFCTICCL=FALSE
NoFCTICT01=FALSE
DemoSucceeded=0100000001
AlternateName=test
OutTableOrder=0100000000000000891F0026CF
OutTableCol0=2C0100002D
OutTableCol1=2C0100002D
AlwaysOnTop=0100000001
NoClop=0000000000
AccessMember=0000000000
```

Siehe auch

[Übersicht Optionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.5 CTI_OGetOptions

Liefert die Optionen als Bitflag zurück.

```
DWORD WINAPI CTI_OGetOptions();
```

Parameter

Keine

Rückgabewert

DWORD

Der Wert enthält Bitflags, welche den Optionen entsprechen.

Bemerkungen

Liefert die [Verbindungsoptionen](#) der Schnittstelle zurück.

Siehe auch

[Übersicht Optionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.6 CTI_OGetOwnerName

Liefert den eigenen Anmeldenamen zurück.

```
PTCHAR WINAPI CTI_OGetOwnerName();
```

Parameter

Keine

Rückgabewert

PTCHAR

Nullterminierte Zeichenkette mit dem Anmeldenamen

Bemerkungen

Der Anmeldenamen entspricht standardmäßig dem Windows-Anmeldenamen. Er kann nachträglich geändert werden.

Siehe auch

[Übersicht Optionen](#) | [CTI_NNetworkConfigDialog](#) | [CTI_OSetOwnerName](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.7 CTI_OGetOwnerNumber

Liefert die eigene Durchwahl zurück.

```
PTCHAR WINAPI CTI_OGetOwnerNumber();
```

Parameter

Keine

Rückgabewert

PTCHAR

Nullterminierte Zeichenkette mit der eigenen Durchwahl

Bemerkungen

Die eigene Durchwahl wird zurückgeliefert. Dieser Text kann zwei bis vier Zeichen lang sein. Ist der Text leer, wurde noch keine interne Durchwahl festgelegt. Dies kann mit [CTI_OSetOwnerNumber](#) geschehen.

Siehe auch

[Übersicht Optionen](#) | [CTI_NNetworkConfigDialog](#) | [CTI_OSetOwnerNumber](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.8 CTI_OGetUserCallInfo

Zeigt, wie der Client über einen neuen Anruf informiert wird.

```
DWORD WINAPI CTI_OGetUserCallInfo();
```

Parameter

Keine

Rückgabewert

DWORD

Zeigt an wie der Client über eingehende Anrufe informiert wird. Mögliche Parameter sind:

0Keine Information

1Der Telefonclient erhält ein Event, worauf er sich nach vorne bringen soll.

2Kundenspezifische Anzeige

Bemerkungen

Die kundenspezifische Anzeige wird im Standardclient verwendet.

Siehe auch

[Übersicht Optionen](#) | [CTI_OSetUserCallInfo](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.9 CTI_OSetDialKey

Legt die Tastenkombination zum Anwählen von markiertem Text fest.

```
VOID WINAPI CTI_OSetDialKey(  
    LONG IValue,  
    DWORD _dwSpecialKey = 2  
);
```

Parameter*IValue*

Virtueller Keycode der F-Taste, kann von 0x70 (F1) bis 0x7B (F12) liegen.

dwSpecialKey

Zusätzliche Taste(n), die gedrückt werden:

0x00 = Keine weitere Taste

0x01 = Umschalttaste

0x02 = Steuertaste

0x03 = Umschalt- und Steuertaste

Rückgabewert

Keiner

Bemerkungen

Ist [CTICLIENT_HOTKEY](#) aktiv geschaltet, so kann markierter Text mit einer F-Taste gewählt werden. Die F-Taste kann mit der Steuertaste und Umschalttaste kombiniert werden. Der Keycode sollte nicht dem gleichen Wert wie unter [CTI_OSetDropKey](#) entsprechen. Die Steuertaste ist Voreinstellung bei den Zusatztasten.

Siehe auch

[Übersicht Optionen](#) | [CTI_OSetOptions](#) | [CTI_OGetDialKey](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.10 CTI_OSetDropKey

Legt die Tastenkombination zum Auflegen von Gesprächen fest.

```
VOID WINAPI CTI_OSetDropKey(  
    LONG IValue,  
    DWORD _dwSpecialKey = 2  
);
```

Parameter*IValue*

Virtueller Keycode der F-Taste, kann von 0x70 (F1) bis 0x7B (F12) liegen.

dwSpecialKey

Zusätzliche Taste(n), die gedrückt werden:

0x00 = Keine weitere Taste

0x01 = Umschalttaste

0x02 = Steuertaste

0x03 = Umschalt- und Steuertaste

Rückgabewert

Keiner

Bemerkungen

Ist [CTICLIENT_HOTKEY](#) aktiv geschaltet, so kann ein Telefongespräch mit einer F-Taste gewählt werden. Die F-Taste kann mit der Steuertaste und Umschalttaste kombiniert werden. Der Keycode sollte nicht dem gleichen Wert wie unter [CTI_OSetDialKey](#) entsprechen.

Siehe auch

[Übersicht Optionen](#) | [CTI_OSetOptions](#) | [CTI_OGetDropKey](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.11 CTI_OSetOptions

Legt die Parameter als Bitflag fest.

```
VOID WINAPI CTI_OSetOptions(  
    DWORD dwAdd,  
    DWORD dwRemove  
);
```

Parameter

dwAdd

Wert, welcher hinzugefügt werden soll.

dwRemove

Flags, welche entfernt werden sollen.

Rückgabewert

Keiner

Bemerkungen

Die Änderungen werden sofort aktiv. Die Zahlenwerte für die Bitflags sind in den [Verbindungsoptionen](#) definiert. Die Einstellungen werden von der Schnittstelle intern gespeichert. Es können gleichzeitig Optionen hinzugefügt und Andere entfernt werden.

Beispiel

[SQL-Demo](#)

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Optionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.12 CTI_OSetOwnerName

Legt den Anmeldenamen fest.

```
VOID WINAPI CTI_OSetOwnerName(  
    PTCHAR pszName
```

```
);
```

Parameter

pszName

Neuer Anmeldename, maximal 31 Zeichen.

Rückgabewert

Keiner

Bemerkungen

Die Anmeldename wird festgelegt. Die CTI-Schnittstelle speichert diesen Wert intern. Der Befehl hat eine Neuanmeldung zur Folge.

Siehe auch

[Übersicht Optionen](#) | [CTI_NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.13 CTI_OSetOwnerNumber

Legt die eigene Durchwahl fest.

```
VOID WINAPI CTI_OSetOwnerNumber(  
    PTCHAR pszNumber  
);
```

Parameter

pszNumber

Eigene Durchwahl, maximal vier Stellen

Rückgabewert

Keiner

Bemerkungen

Die eigene Durchwahl wird festgelegt. Diese kann zwei bis vier Stellen haben. Die CTI-Schnittstelle speichert diesen Wert intern. Damit sich der Client unter der neuen Durchwahl am CTI-Server anmeldet, rufen Sie [CTI_NClientRestart](#) auf.

Siehe auch

[Übersicht Optionen](#) | [CTI_NNetworkConfigDialog](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.2.14 CTI_OSetUserCallInfo

Legt fest, wie der Benutzer bei eingehenden Anrufen informiert wird.

```
VOID WINAPI CTI_OSetUserCallInfo(  
    DWORD dwUserCallInfo  
);
```

Parameter*dwUserCallInfo*

Legt fest wie der an der Schnittstelle angedockte Client über einen eingehenden Anruf informiert werden soll. Mögliche Werte sind:

0 = Keine Information

1 = Der Telefonclient erhält ein Event, worauf er sich nach vorne bringen soll.

2 = Kundenspezifische Anzeige

Rückgabewert

Keiner

Bemerkungen

Die kundenspezifische Anzeige wird im Standardclient verwendet.

Siehe auch

[Übersicht Optionen](#) | [CTI_CALLNUMBER](#) | [CTI_OGetUserCallInfo](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3 Telefoniefunktionen

Verwendung der Telefoniefunktionen

Die TAPIMaster®-Schnittstelle kapselt die TAPI-Schnittstelle und reduziert deren Befehlsumfang deutlich. Die auf dieser Seite aufgeführten Funktionen dienen der Steuerung des Telefons oder sind verwandter Natur. Nicht alle Funktionen sind auf allen Anlagen möglich, dies hängt von der Leistungsfähigkeit des jeweiligen Treibers ab.

In der TAPIMaster®-Schnittstelle hat jeder Anruf eine eindeutige Nummer (Hat nichts mit der Rufnummer zu tun), mit der die Anrufe unterschieden werden können. Sie sollten sich diese Nummern intern merken, da sie bei vielen Telefoniebefehlen gebraucht werden. Sie wird als Anruf-ID bezeichnet.

Präfix

Die Telefoniefunktionen haben den Präfix "CTI_T" vorangestellt.

Single Call Funktionen

CTI_TMakeCall	Startet einen Anruf.
CTI_TCallAnswer	Nimmt einen Anruf entgegen.
CTI_TCallDrop	Legt einen Anruf auf.

Multicall Funktionen

CTI_TAddCallToConference	Fügt einen bestehenden Anruf einer Konferenz hinzu.
CTI_TAddNumberToConference	Ruft einen weiteren Teilnehmer an und holt diesen in die Konferenz.
CTI_TCallChange	Veraltet. Benutzen Sie CTI_TCallToggle .
CTI_TCallHold	Ein Anruf wird gehalten.
CTI_T_CallRedirect	Ein Anruf wird weitergereicht.
CTI_TCallRedirection	Veraltet. Benutzen Sie CTI_T_CallRedirect .
CTI_TCallToggle	Schaltet zwischen aktiven und gehaltenem Anruf hin und her.
CTI_TCallTransfer	Zwei Anrufe werden verbunden.
CTI_TCallUnHold	Holt einen gehaltenen Anruf zurück.
CTI_TPickup	Holt einen Anruf von einem anderen Teilnehmer heran.
CTI_TProxyMakeCall	Ein anderes Telefon ruft Teilnehmer an, tätigt Rückfrage zum Ziel und verbindet.

Hilfsfunktionen

CTI_TLineReset	TAPI-Leitung wird zurückgesetzt.
CTI_TReDial	Startet eine Wahlwiederholung.

CTI_TRemovePhonelcon	Das Anrufsymbol wird entfernt.
CTI_TSendCallReminder	Eine Rückruferinnerung wird gesendet.
CTI_TSendRecall	Veraltet. Benutzen Sie CTI_TSendCallReminder .
CTI_TGetForwardingState	Liefert den Status der Rufumleitung zurück.
CTI_TGetRedirectionState	Veraltet. Benutzen Sie CTI_TGetForwardingState .
CTI_TGetForwardingDest	Liefert das Rufumleitungsziel zurück.
CTI_TGetRedirectionNumber	Veraltet. Benutzen Sie CTI_TGetForwardingDest .
CTI_TSetForwarding	Die Rufumleitung wird zu- oder abgeschaltet.
CTI_TSetRedirection	Veraltet. Benutzen Sie CTI_TSetForwarding .

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.1 CTI_TAddCallToConference

Fügt ein Gespräch zu einer Konferenz hinzu.

```
VOID WINAPI CTI_TAddCallToConference(
    LONG ICallID = 0
);
```

Parameter

ICallID

Anruf-ID des Gesprächs, welches zu einer Konferenz hinzugefügt werden soll. Wenn hier 0 steht dann versucht die CTI-Engine automatisch entweder eine Konferenz aufzubauen oder den wahrscheinlichsten Teilnehmer hinzuzufügen.

Rückgabewert

Keiner

Bemerkungen

Mit dieser Funktion können Sie bereits bestehende Gespräche zu einer Konferenz hinzufügen. In der Regel werden Sie ein gehaltenes Gespräch nach einer Rückfrage mit in das aktive Gespräch holen. Nicht alle Anlagen unterstützen diese Funktion. Insbesondere Konferenzen mit mehr als drei Teilnehmern werden nur von wenigen Herstellern unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TAddNumberToConference](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.2 CTI_TAddNumberToConference

Tätigt einen Anruf und holt diesen in die Konferenz.

```
VOID WINAPI CTI_TAddNumberToConference(  
    PTCHAR pszNumber  
);
```

Parameter

pszNumber

Rufnummer des Teilnehmers, welcher angerufen und in die Konferenz geholt werden soll.

Rückgabewert

Keiner

Bemerkungen

Die Funktion ruft *pszNumber* an. Nimmt dieser das Gespräch entgegen, so wird er in die Konferenz mit einbezogen bzw. das aktive Gespräch zu einer Dreierkonferenz ausgebaut. Nicht alle Anlagen unterstützen diese Funktion. Insbesondere Konferenzen mit mehr als drei Teilnehmern werden nur von wenigen Herstellern unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TAddCallToConference](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.3 CTI_TCallAnswer

Nimmt ein Gespräch entgegen.

```
VOID WINAPI CTI_TCallAnswer(  
    LONG ICallID = 0  
);
```

Parameter

ICallID

Anruf-ID des Gesprächs, das entgegengenommen werden soll. Wird 0 übergeben, so wird versucht den ersten Anruf in der eigenen Warteschlange entgegen zunehmen, der sich im State [OFFERING](#) befindet.

Rückgabewert

Keiner

Bemerkungen

Nimmt einen eingehenden Anruf entgegen. Die Funktion hat nur dann Erfolg, wenn die Telefone ins Freisprechen schalten können, die meisten ISDN- und Systemtelefone können dies.

Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_CALLSTATE](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.4 CTI_TCallDrop

Ein Anruf wird aufgelegt.

```
VOID WINAPI CTI_TCallDrop(  
    LONG ICallID  
);
```

Parameter

ICallID

Anruf-ID des Gesprächs, welches aufgelegt werden soll. Wird 0 übergeben, so wird das erste Gespräch in der Leitung aufgelegt.

Rückgabewert

Keiner

Bemerkungen

Legt ein Gespräch auf. Ist in fast allen Treibern implementiert.

Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TMakeCall](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.5 CTI_TCallHold

Ein Gespräch wird gehalten.

```
VOID WINAPI CTI_TCallHold(  
    LONG ICallID = 0  
);
```

Parameter

ICallID

Anruf-ID des Gesprächs, das ins Halten gestellt werden soll. Wenn diese 0 ist dann wird der wahrscheinlichste Anruf gehalten.

Rückgabewert

Keiner

Bemerkungen

Stellt einen aktiven Anruf in den gehaltenen Zustand. Der Anrufer hört Wartemusik. Mit [CTI_TCallUnHold](#) kann der Anruf zurückgeholt werden. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TCallToggle](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.6 CTI_TCallRedirect

Ein Gespräch wird weitergeleitet.

```
VOID WINAPI CTI_TCallRedirect(
    LONG ICallID = 0,
    PTCHAR pszNumber
);
```

Parameter

ICallID

Anruf-ID des Gesprächs, das umgeleitet werden soll.

pszNumber

Zielnummer, zu der ein Anruf umgeleitet werden soll.

Rückgabewert

Keiner

Bemerkungen

Die Funktion leitet Anrufe zu anderen Teilnehmern weiter. Es ist gleichgültig, ob der Anruf bereits verbunden ist oder noch nicht entgegengenommen wurde. Bei guten Anlagen können auch anklopfende Gespräche weitergeleitet werden. Wenn ICallID Null ist dann wird der wahrscheinlichste Anruf im State OFFERING / ACCEPTED weitergeleitet. Ist kein Anruf in dem State dann ist das wahrscheinlichste verbundene Gespräch Gegenstand der Weiterleitung.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.7 CTI_TCallToggle

Wechselt zwischen zwei Anrufen hin und her.

```
VOID WINAPI CTI_TCallToggle(
    LONG IOldCall = 0,
    LONG INewCall = 0
);
```

Parameter*IOldCall*

Anruf-ID des Gesprächs, das gerade verbunden ist und ins Halten gestellt werden soll.

INewCall

Anruf-ID des Gesprächs, welches gerade gehalten wird und verbunden werden soll.

Rückgabewert

Keiner

Bemerkungen

Die Funktion wechselt zwischen zwei Anrufen hin und her. Ein Gespräch ist jeweils verbunden, das andere Gespräch wird gehalten. Die Parameter können auch vertauscht sein, die Funktion ist fehlertolerant. Sind die Parameter 0 dann Sucht die CTI-Engine automatisch nach den beiden zu tauschenden Anrufen. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TCallHold](#) | [CTI_TCallUnHold](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.8 CTI_TCallTransfer

Zwei Anrufe werden verbunden.

```
VOID WINAPI CTI_TCallTransfer(  
    LONG ICall1 = 0,  
    LONG ICall2 = 0  
);
```

Parameter*ICall1*

Anruf-ID des ersten Anrufs

ICall2

Anruf-ID des zweiten Anrufs

Rückgabewert

Keiner

Bemerkungen

Zwei Anrufe sollen miteinander verbunden werden. Meist ist ein Gespräch davon aktiv und das andere gehalten (hört Wartemusik). Die Reihenfolge in der die Anruf-IDs übergeben werden spielt keine Rolle. Die Funktion wird nicht von allen Telefonanlagen unterstützt. Ist einer oder beide Parameter 0 dann sucht die CTI-Engine automatisch nach den beiden Anrufen und verbindet die mit der größten Wahrscheinlichkeit.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.9 CTI_TCallUnHold

Ein gehaltenes Gespräch wird zurückgeholt.

```
VOID WINAPI CTI_TCallUnHold(  
    LONG ICallID = 0  
);
```

Parameter

ICallID

Anruf-ID des Gesprächs, welches wieder verbunden werden soll. Ist der Parameter 0 dann wird das letzte gehaltene Gespräch zurückgeholt.

Rückgabewert

Keiner

Bemerkungen

Ein mit [CTI_TCallHold](#) gehaltener Anruf kann zurückgeholt werden, so dass das Gespräch wieder aktiv ist. Die Funktion wird nicht von allen Telefonanlagen unterstützt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TCallToggle](#) | [CTI_TCallHold](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.10 CTI_TGetForwardingDest

Das Rufumleitungsziel wird ausgegeben.

```
PTCHAR WINAPI CTI_TGetForwardingDest();
```

Parameter

Keine

Rückgabewert

PTCHAR

Ziel der Rufumleitung als Zeichenkette

Bemerkungen

Die Rufumleitung muss nicht aktiv sein, auch wenn eine Nummer zurückgeliefert wird. Mit [CTI_TGetForwardingState](#) können Sie feststellen, ob die Umleitung aktiv ist.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TSetForwarding](#) | [CTI_TGetForwardingState](#) | [CTI_FORWARDINGSTATE](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.11 CTI_TGetForwardingState

Der Status der Rufumleitung wird ausgegeben.

```
BOOL WINAPI CTI_TGetForwardingState();
```

Parameter

Keine

Rückgabewert

BOOL

Ist TRUE wenn eine Umleitung besteht.

Bemerkungen

Zeigt an ob eine Rufumleitung besteht. Mit [CTI_TGetForwardingDest](#) können Sie das Ziel einer Rufumleitung abfragen.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TGetForwardingDest](#) | [CTI_TSetForwarding](#) | [CTI_FORWARDINGSTATE](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.12 CTI_TLineReset

Die TAPI-Leitung wird zurückgesetzt.

```
VOID WINAPI CTI_TLineReset();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Mitunter kann es vorkommen, dass der TAPI-Treiber nicht mehr reagiert. Diese Funktion setzt die Leitung zurück, in den meisten Fällen sollte dann das Problem behoben sein. Sie haben auch die Möglichkeit, im CTI-Server eine Autokorrektur zuzuschalten, welche versucht solche Probleme zu beheben. In der Serverkonsole finden Sie auf der Seite "TAPI-Sonderbehandlung" dazu die Möglichkeit.

**Siehe auch**

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.13 CTI_TMakeCall

Ein Anruf wird gestartet.

```
VOID WINAPI CTI_TMakeCall(  
    PTCHAR pszNumber  
);
```

Parameter

pszNumber
Rufnummer der angerufenen Stelle

Rückgabewert

Keiner

Bemerkungen

Sollte bereits ein aktives Gespräch bestehen, wird versucht dieses ins Halten zu stellen und einen neuen Anruf zu tätigen. Sie können diese Funktion auch für Rückfragen einsetzen.

Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TCallDrop](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.14 CTI_TPickup

Ein eingehendes, nicht entgegengenommenes Gespräch wird herangeholt.

```
VOID WINAPI CTI_T_Pickup(  
    PTCHAR pszPickupNumber  
);
```

Parameter

pszPickupNumber
Durchwahl des Apparats, von dem ein Gespräch herangeholt werden soll.

Rückgabewert

Keiner

Bemerkungen

Mit Hilfe dieser Funktion lassen sich Gespräche von anderen Apparaten heran holen. TAPIMaster® arbeitet hier teilweise mit Ersatzfunktionen gegenüber der TAPI, so dass auch ein anlagenübergreifendes Heranholen von Gesprächen möglich ist. Sollte die Funktion trotzdem fehlschlagen, kann TAPIMaster® direkt Hardware-Steuerbefehle absetzen um das Gespräch heranzuholen. Je nach Anlage können auch anklopfende Gespräche gepickt werden.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.15 CTI_TProxyMakeCall

Ein Anruf wird über ein Proxytelefon gestartet. Nützlich bei einigen DECT Telefonen.

```
VOID WINAPI CTI_TProxyMakeCall(  
    PTCHAR pszNumber,  
    PTCHAR pszProxy,  
);
```

Parameter

pszNumber

Rufnummer der angerufenen Stelle.

pszProxy

Durchwahl des vermittelnden Telefons, muss ein Systemtelefon sein.

Rückgabewert

Keiner

Bemerkungen

Ein Teilnehmer kann nicht selbst über TAPI telefonieren. Ein Vermittlertelefon ruft den Akteur an. Der Akteur hebt ab. Daraufhin startet das Vermittlertelefon eine Rückfrage zum eigentlichen Anrufziel. Das Gespräch wird noch in der Anwahl übergeben.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TCallDrop](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.16 CTI_TReDial

Die Wahlwiederholung wird gestartet.

```
VOID WINAPI CTI_TReDial();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Tätigt eine Wahlwiederholung der zuletzt angerufenen Nummer.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.17 CTI_TRemovePhoneIcon

Das Telefonsymbol wird entfernt.

```
VOID WINAPI CTI_TRemovePhoneIcon();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Bei eingehenden Anrufen kann ein kleines Telefonsymbol eingeblendet sein, wenn die Option [CTIClient_PHONEICON](#) aktiviert wurde. Mit dieser Funktion kann es wieder entfernt werden wenn das Telefonat bearbeitet ist.

**Siehe auch**

[Übersicht Telefoniefunktionen](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.18 CTI_TSendCallReminder

Eine Anruferinnerung wird gesendet.

```
VOID WINAPI CTI_TSendCallReminder(  
    PTCHAR pszOther,  
    PTCHAR pszNumber,  
    PTCHAR pszInfo = NULL  
);
```

Parameter

pszOther

Durchwahl des Teilnehmers, welcher informiert werden soll.

pszNumber

Rufnummer des Anrufers, der zurückgerufen werden soll.

pszInfo

Kurze Nachricht (bis 255 Zeichen), die das Problem des Anrufers beschreiben.

Rückgabewert

Keiner

Bemerkungen

Ein Anruf kommt herein. Es stellt sich heraus dass der Anrufer einen Kollegen sprechen wollte der aber gerade nicht da ist. Mit dieser Funktion kann der Kollege schnell und einfach über den Anruf informiert werden. Es wird ein Event vom Typ [CTI_CALLREMINDER](#) im Client des Kollegen ausgelöst. Ist dieser nicht angemeldet, erhält er das Event bei seiner Anmeldung. Somit gehen weniger Informationen verloren, die Daten können auf dem Zielclient gleich für eine Anrufvorlage genutzt werden, zeitraubendes Abschicken von Mails entfällt.

Siehe auch

[Übersicht Telefoniefunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.3.19 CTI_TSetForwarding

Die Rufumleitung wird zu- oder abgeschaltet.

```
VOID WINAPI CTI_TSetForwarding(  
    BOOL bSet,  
    PTCHAR pszNumber = _T("")  
);
```

Parameter

bSet

Zu- oder Abschalten der Umleitung

pszNumber

Nummer, zu der umgeleitet werden soll.

Rückgabewert

Keiner

Bemerkungen

Schaltet eine Rufumleitung oder zu oder ab. Als Antwort darauf erhält der Teilnehmer ein Event vom Typ [CTI_FORWARDINGSTATE](#). Die Funktion kann fehlschlagen, wenn die Zielnummer bereits umgeleitet hat, die Rechte zur Umleitung nicht vorliegen oder die TAPI-Schnittstelle der Anlage die Funktion nicht unterstützt. Im letzten Fall gibt es die Möglichkeit, die Rufumleitung über Telefoncodes zu steuern, diese werden in der Serverkonsole auf der Seite "TAPI-Sonderbehandlungen" eingestellt.

Siehe auch

[Übersicht Telefoniefunktionen](#) | [CTI_TGetForwardingState](#) | [CTI_TGetForwardingDest](#) | [CTI_FORWARDINGSTATE](#)

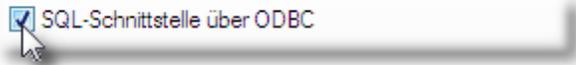
[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.4 SQL

Verwendung der SQL-Funktionen

In Telefonieanwendungen kann es erforderlich sein, auf Datenbankbestände zuzugreifen um z. B. zu einer Rufnummer weitere Daten wie Adresse, Notizen usw. anzeigen zu können. Der CTI-Server unterhält bereits Datenbankverbindungen um Rufnummern identifizieren zu können oder Gesprächsdatensätze zu speichern. Mit einer dritten Datenbankverbindung haben Sie die Möglichkeit, individuelle Abfragen zu tätigen. Richten Sie dazu erst die Datenquelle auf dem CTI-Server ein. Auf der Seite "Datenbankschnittstelle" der Serverkonsole finden Sie die entsprechenden Einstelloptionen.



Danach empfiehlt es sich die Befehle [CTI_SQLLogin](#), [CTI_SQLBegin](#), [CTI_SQLNext](#) und [CTI_SQLEnd](#) zuerst einmal mit dem [CTI-Browser](#) zu testen.

Präfix

Die SQL-Funktionen haben den Präfix "SQL_" vorangestellt.

Adressen aus der Stammdatenbank holen

Aus der Datenbank für die Rufnummernidentifikation lassen sich auch umgekehrt Rufnummern zu Namen ermitteln.

CTI_SQLGetDbBo ok	Holt einzelne Datensätze oder eine komplette Adreßliste aus der Stammdatenbank.
--	---

Abfragen mit zusätzlicher Datenbankverbindung

Nutzen Sie diese Befehle für individuelle Datenbankabfragen.

CTI_SQLLogin	Meldet Client bei der Datenbank an.
CTI_SQLBegin	Startet eine Datenbankabfrage.
CTI_SQLNext	Setzt eine Abfrage fort.
CTI_SQLEnd	Beendet eine Abfrage.

Möglicher Ablauf einer Sitzung

1. Anmelden an Datenbank mit [CTI_SQLLogin](#)("Paßwort");
2. [SQL_LOGIN](#) liefert TRUE in wParam zurück, Anmeldung erfolgreich
3. 50 Datensätze anfordern mit [CTI_SQLBegin](#)("SELECT * FROM members",50); Funktion liefert Request von 1 zurück.
4. Datensätze kommen mit Event [SQL_DATA](#) an.
5. Weitere 30 Datensätze werden angefordert für Abfrage Nummer 1: [CTI_SQLNext](#)(1,30);
6. Datensätze kommen mit Event [SQL_DATA](#) an.
7. Abfrage Nummer 1 beenden mit [CTI_SQLEnd](#)(1);
8. Das Event [SQL_END](#) bestätigt das Ende der Abfrage.

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.4.1 CTI_SQLBegin

Eine SQL-Abfrage wird gestartet.

```
DWORD WINAPI CTI_SQLBegin(  
    PTCHAR pszCommand,  
    DWORD dwMaxCount  
);
```

Parameter

pszCommand

SQL-Kommando, welches an die Datenbank geschickt wird.

dwMaxCount

Maximale Anzahl der Datensätze, die abgefragt werden soll.

Rückgabewert

DWORD

Abfrage-ID

Bemerkungen

Die Funktion startet eine Datenbankabfrage. Es kann ein beliebiger SQL-String übergeben werden, sofern der Administrator hier keine Beschränkungen vorgenommen hat. Da die Zahl der ermittelten Datensätze sehr groß werden kann sollte sie mit *dwMaxCount* sinnvoll beschränkt werden. Die Abfrage lässt sich mit [CTI_SQLNext](#) fortsetzen und [CTI_SQLEnd](#) beenden. Mit **CTI_SQLBegin** gestartete Abfragen werden nach einigen Minuten selbständig auf dem Server terminiert. Bei jeder Abfrage wird eine Abfrage-ID zurückgeliefert. Diese wird beim Aufruf von [CTI_SQLNext](#) und [CTI_SQLEnd](#) benötigt. Wird 0 zurückgeliefert so konnte die Abfrage nicht gestartet werden. Das ist der Fall wenn der Benutzer nicht an der Datenbank angemeldet ist.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [SQL_BEGIN](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.4.2 CTI_SQLEnd

Eine SQL-Abfrage wird beendet.

```
VOID WINAPI CTI_SQLEnd(  
    DWORD dwRequestID  
);
```

Parameter

dwRequestID

Abfrage-ID, welche mit [CTI_SQLBegin](#) zurückgeliefert wurde.

Rückgabewert

Keiner

Bemerkungen

Eine Abfrage wird beendet. Daraufhin erhält der Client einen Event vom Typ [SQL_END](#). Da für jede Abfrage ein Statementhandle im Datenbankzugriff angelegt wird, sollten nicht benötigte Abfragen so schnell wie möglich terminiert werden. Nicht geschlossene Abfragen werden nach einigen Minuten automatisch beendet.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [SQL_END](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.4.3 CTI_SQLGetDbBook

Rufnummerdatensätze zu Namen oder Namensbestandteilen werden gesucht.

```
VOID WINAPI CTI_SQLGetDbBook(  
    PTCHAR pszText  
);
```

Parameter

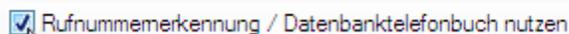
pszText
Suchkriterium

Rückgabewert

Keiner

Bemerkungen

Die Funktion sucht in der Datenbank für die Rufnummernidentifikation nach Rufnummern. Hier sind Namen oder Namensfragmente gegeben und Datensätze mit Namen und Rufnummern werden zurückgeliefert. Ein "Sch" zum Beispiel sucht nach allen Namen wie Schmidt oder Schulz. Wird ein Leerstring übergeben, so erstellt die Datenbank ein komplettes Telefonbuch und schickt es an den Client zurück. Die zurückgesendeten Daten lösen im Client einen Event vom Typ [SQL_DBBOOK](#) aus. Bevor diese Funktion genutzt werden kann ist es erforderlich dass die Datenbankschnittstelle des CTI-Servers eingerichtet wurde. Die Rufnummernerkennung auf der Seite "Datenbankschnittstelle" der Serverkonsole sollte aktiviert und eingerichtet sein.



Rufnummernerkennung / Datenbanktelefonbuch nutzen

Siehe auch

[Übersicht SQL](#) | [SQL_DBBOOK](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.4.4 CTI_SQLLogin

Die Anmeldung an die Datenbank erfolgt.

```
VOID WINAPI CTI_SQLLogin(  
    PTCHAR pszPassword  
);
```

Parameter

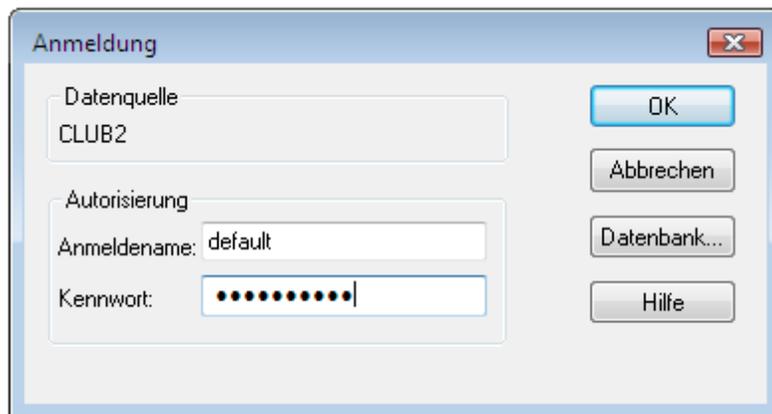
pszPassword
Paßwort der Datenbank

Rückgabewert

Keiner

Bemerkungen

Um die SQL-Schnittstelle nutzen zu können, meldet sich der Client mit dem Paßwort der Datenbank an. Das Paßwort wurde vom Administrator beim Einrichten der ODBC-Datenquelle vergeben.



pszPassword kann auch ein leerer String sein, wenn kein Paßwort vergeben wurde. Als Antwort darauf erhält der Client einen Event vom Typ [SQL_LOGIN](#).

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [SQL_LOGIN](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.4.5 CTI_SQLNext

SQL-Abfrage wird fortgesetzt.

```
BOOL WINAPI CTI_SQLNext(  
    DWORD dwRequestID,  
    DWORD dwMaxNextCount  
);
```

Parameter

dwRequestID
Abfrage-ID

dwMaxNextCount
Maximale Anzahl von Datensätzen

Rückgabewert

BOOL
TRUE, wenn Abfrage gesendet werden konnte.

Bemerkungen

Eine mit [CTI_SQLBegin](#) gestartete Abfrage wird fortgesetzt. Dazu wird die Abfrage-ID übergeben, welche [CTI_SQLBegin](#) zurückgeliefert hat. Die Anzahl der maximal zu liefernden Datensätze wird begrenzt auf so viele Datensätze, wie in nächster Zeit sinnvoll weiterverarbeitet werden kann. Die Lebensdauer einer Abfrage verlängert sich durch jedes **CTI_SQLNext**. Ist der Client nicht an die Datenbank angemeldet, so schlägt die Funktion fehl und liefert FALSE zurück.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht SQL](#) | [SQL_DATA](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5 Zusatzfunktionen

Verwendung der Zusatzfunktionen

Die CTI-Schnittstelle wurde um einige nützliche Hilfsfunktionen ergänzt, welche zwar meist nichts mit Telefonie zu tun haben aber die Programme bereichern können. So wird Chat und das Senden von Nachrichten unterstützt.

Präfix

Die Zusatzfunktionen haben den Präfix "CTI_E" vorangestellt.

Nachrichten senden

CTI_EChatStart	Startet eine Chatsitzung.
CTI_EChatMessage	Die Chatnachricht wird übertragen.
CTI_EChatEnd	Die aktive Chatsitzung wird beendet.
CTI_ESendUserToUser	Eine Kurznachricht wird geschickt.
CTI_EUserSendToDataBase	Sendet benutzerdefinierte Nachrichten.

Sonstige

CTI_EExpandInterNumber	Expandiert eine Durchwahl.
CTI_EGetDateTime	Wandelt time_t in Standardstring um.
CTI_EGetFormatDateTime	Wandelt time_t in individuell formatierten String um.
CTI_ESetAcidReady	Bereitschaft zum Gesprächsempfang in der ACD-Gruppe signalisieren.
CTI_ESetMainWindow	Übergibt das Hauptfensterhandle an die Schnittstelle.
CTI_EShowInfoDialog	Zeigt Versionsinfo der Schnittstelle an.
CTI_EConvertToCanonical	Wandelt eine Rufnummer in das kanonische Format um.

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.1 CTI_EChatEnd

Die aktive Chatsitzung wird beendet.

```
VOID WINAPI CTI_EChatEnd();
```

Parameter

Keine

Rückgabewert

Keiner

Bemerkungen

Die aktive Chatsitzung wird beendet. Jeder der beiden Chatpartner kann diese Funktion aufrufen. Darauf hin erhalten beide Seiten einen Event vom Typ [CTI_CHATSTOP](#). Die Anwendung kann dann die Chatfenster schließen oder deaktivieren.

Siehe auch

[Übersicht Zusatzfunktionen](#) | [CTI_EChatStart](#) | [CTI_EChatMessage](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.2 CTI_EChatMessage

Die Chatnachricht wird übertragen.

```
VOID WINAPI CTI_EChatMessage(  
    PTCHAR pszMsg  
);
```

Parameter

pszMsg

Nachrichtentext, der in der Chatbox steht.

Rückgabewert

Keiner

Bemerkungen

Es wird der komplette Text in der Chatbox übertragen. Die Funktion sollte aufgerufen werden sobald sich im jeweiligen Eingabefeld etwas ändert. Die Funktion ruft beim Chatpartner den Event [CTI_CHATMESSAGE](#) auf.

Siehe auch

[Übersicht Zusatzfunktionen](#) | [CTI_EChatStart](#) | [CTI_EChatEnd](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.3 CTI_EChatStart

Startet eine Chatsitzung.

```
VOID WINAPI CTI_EChatStart(  
    PTCHAR pszDest  
);
```

Parameter

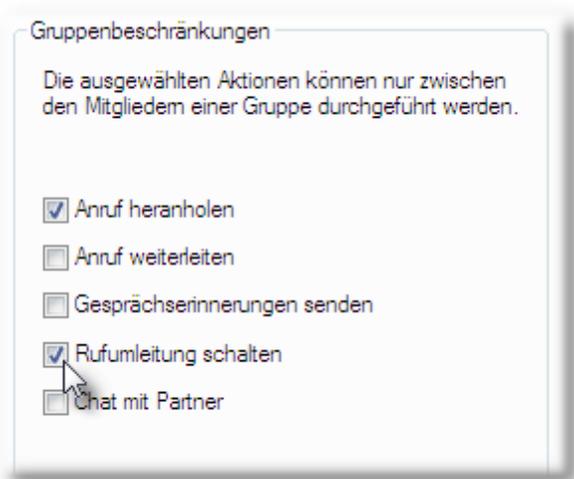
pszDest
Durchwahl des Chatpartners

Rückgabewert

Keiner

Bemerkungen

Startet eine Chatsitzung. Der Chatpartner muss angemeldet und die Rechte zum Chat vorhanden sein; es darf seinerseits noch keine Chatsitzung geben. Wenn die Anmeldung zum Chat zulässig war, erhalten beide Partner einen Event vom Typ [CTI_CHATSTART](#). Daraufhin kann die Anwendung Ein- und Ausgabefelder für die Chatsitzung einblenden. Sollte Chat nicht funktionieren könnte es daran liegen dass Chat auf die Mitglieder der gleichen Gruppe beschränkt ist. Auf der Seite "Gruppenverwaltung" der Serverkonsole können Sie diese Beschränkungen aufheben.

**Siehe auch**

[Übersicht Zusatzfunktionen](#) | [CTI_EChatMessage](#) | [CTI_EChatEnd](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.4 CTI_EConvertToCanonical

Umwandlung von Rufnummern in das kanonische Format.

```
VOID WINAPI CTI_EConvertToCanonical(  
    PTCHAR pszInput,  
    PTCHAR pszOutput  
)
```

Parameter

pszInput
Rufnummer, die in das kanonische Format umgewandelt werden soll.

pszOutput
Umgewandelte Rufnummer

Bemerkungen

Sie können diese Funktion benutzen um Datenbestände mit Rufnummern in das kanonische Format umzuwandeln. Der Ausgabestring sollte mindestens 32 Zeichen groß sein.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.5 CTI_EExpandInternNumber

Expandiert eine Durchwahl.

```
PTCHAR WINAPI CTI_EExpandInternNumber(
    PTCHAR pszNumber
);
```

Parameter

pszNumber
Interne Rufnummer, welche erweitert werden soll.

Rückgabewert

PTCHAR
Liefert die erweiterte Nummer als nullterminierten String zurück.

Bemerkungen

Erweitert eine Rufnummer um die Landeskenntung, Vorwahl und ggf. Anlagenrufnummer.

Beispiel

Bei diesen Einstellungen liefert CTI_EExpandInternNumber("22") in Deutschland den Wert +49 (6033) 44422.

Amt	Int...	Filter ...	Filter ...	Vorwahl	Anlagennummer
...	0	0	0	6033	444

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.6 CTI_EGetDateTime

Wandelt time_t in Standardstring um.

```
PTCHAR CTI_EGetDateTime(
    __time32_t wTime,
);
```

Parameter

wTime

Wert im Format time_t, welcher z. B. im Event [CTI_PROTOCOLDATA](#) zurückgeliefert wird.

Rückgabewert

Nullterminierter String in einem Standardformat: 2007-01-12T00:53:32.875

Bemerkungen

Nutzen Sie diese Funktion wenn Sie ein gut sortierbares Format brauchen. Viele Programme wie Excel nutzen dieses Format.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.7 CTI_EGetFormatDateTime

Wandelt time_t in individuell formatierten String um.

```
PTCHAR CTI_EGetFormatDateTime(
    __time32_t wTime,
    PTCHAR pszFormat
);
```

Parameter

wTime

Wert im Format time_t, welcher z. B. im Event [CTI_PROTOCOLDATA](#) zurückgeliefert wird.

pszFormat

String mit Datumsformat. Dieses kann aus den folgenden Formatzeichen gebildet werden.

Zeichen	Erklärung
%a	Abgekürzter Wochentag
%A	Ausgeschriebener Wochentag
%b	Abgekürzter Monatsname
%B	Ausgeschriebener Monatsname
%c	Lokales Datum und Uhrzeit, abhängig von Systemeinstellungen
%d	Monatstag (01 – 31)
%H	Stunde im 24Stunden-Format (00 – 23)
%I	Stunde im 12Stunden-Format (01 – 12)

%j	Tag im Jahr (001 – 366)
%m	Monat als Zahl (01 – 12)
%M	Minute (00 – 59)
%p	Lokaler Indikator für vormittag / nachmittag bei 12stündiger Anzeige
%S	Sekunden (00 – 59)
%U	Kalenderwoche, beginnend mit Sonntag (00 – 53)
%w	Wochentag als Zahl, beginnend mit Sonntag (0 – 6)
%W	Kalenderwoche, beginnend mit Montag (00 – 53)
%x	Lokales Datum, abhängig vom System
%X	Lokale Zeit, abhängig vom System
%y	2stellige Jahresangabe (00 – 99)
%Y	4stellige Jahresangabe
%z,%Z	Zeitzone
%%	Prozentzeichen

Rückgabewert

Nullterminierter String in einem individuellen Format.

Bemerkungen

Die Formatzeichen können beliebig kombiniert werden.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.8 CTI_ESendUserToUser

Eine Kurznachricht wird geschickt.

```
VOID WINAPI CTI_ESendUserToUser(
    PTCHAR pszOtherUser,
    PTCHAR pszMessage
);
```

Parameter

pszOtherUser

Durchwahl des anderen Teilnehmers

pszMessage

Kurznachricht (max 220 Zeichen), die an ihn geschickt wird.

Rückgabewert

Keiner

Bemerkungen

Verschiedene Benutzer können sich mit dieser Funktion Kurznachrichten schicken, ohne den Mailclient bemühen zu müssen. Auf dem Client des Empfängers wird der Event [CTI_USERTOUSER](#) ausgelöst.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.9 CTI_ESetAcidReady

Der Client der Servicegruppe teilt dem Server seine Bereitschaft zum Gesprächsempfang mit.

```
VOID WINAPI CTI_ESetAcidReady(  
    BOOL bReady,  
);
```

Parameter

bReady

Ist TRUE wenn der Client für die Hotline zur Verfügung stehen soll.

Rückgabewert

Keiner

Bemerkungen

Der Client muss Mitglied der Servicegruppe sein.

Siehe auch

[Übersicht Zusatzfunktionen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.10 CTI_ESetMainWindow

Übergibt das Hauptfensterhandle an die Schnittstelle.

```
VOID WINAPI CTI_ESetMainWindow(  
    HWND hWnd,  
);
```

Parameter

hWnd

Fensterhandle der Anwendung

Rückgabewert

Keiner

Bemerkungen

Übergibt der Schnittstelle das Handle des Hauptfensters der Anwendung. Wird benötigt, um Unterdialoge der Schnittstelle an der richtigen Position anzuzeigen.

Beispiel[SQL-Demo](#)[Mini CTI Client \(C++\)](#)**Siehe auch**[Übersicht Zusatzfunktionen](#) | [CTI_EShowInfoDialog](#) | [CTI_NNetworkConfigDialog](#)[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.11 CTI_EShow InfoDialog

Zeigt Versionsinfo der Schnittstelle an.

VOID WINAPI CTI_EShowInfoDialog();**Parameter**

Keine

Rückgabewert

Keiner

Bemerkungen

Zeigt einen kleinen Dialog mit Informationen zur Schnittstelle an.

**Siehe auch**[Übersicht Zusatzfunktionen](#)[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.5.12 CTI_EUserSendToDataBase

Sendet benutzerdefinierte Nachrichten.

**VOID WINAPI CTI_EUserSendToDataBase(
LONG ICommand,**

PTCHAR pszMessage

);

Parameter*ICommand*

Nummer des Befehls

pszMessage

Übermittelte Parameter

Rückgabewert

Keiner

Bemerkungen

Mit dieser Funktion können benutzerdefinierte Nachrichten an die Serverschnittstelle geschickt werden. Dort lösen diese einen Event vom Typ [CTI_USERTODATABASE](#) aus. Anhand von *ICommand* werden die dort eingehenden Befehle unterschieden. Die Serverschnittstelle kann auf die Nachricht antworten und löst auf der Clientseite den Event [CTI_DATABASETOUSER](#) aus.

Siehe auch

[Übersicht Zusatzfunktionen](#) | [C++ CTI API \(Server\)](#) | [CTI_DATABASETOUSER](#) | [CTI_USERTODATABASE](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.6 Events

Verwendung der Events

Die Events der Clientschnittstelle können entweder als Windowsnachricht oder über eine Callbackfunktion kommen, je nach Initialisierung der Schnittstelle. Die Callbackfunktion hat einen ähnlichen Aufbau wie die Windowsfunktion, lediglich das Fensterhandle fehlt:

```
typedef VOID WINAPI CallbackFunc(DWORD, WPARAM, LPARAM);
```

Handelt es sich bei der Callbackfunktion um eine Objektfunktion, so muss diese statisch sein.

Typen

Im DWORD werden dann auch Windowsnachrichten übertragen, welche unten aufgeführt sind.

Typ	Wert	Beschreibung
WM_CTI_MENU OMMAND	WM_USER + 31	Nachricht vom Kontextmenü des Verbindungssymbols
WM_CTI_SHOW PHONELIST	WM_USER + 32	Nachricht vom Anrufsymbol
WM_CTI_NETWO RKCLOSE	WM_USER + 23	Netzwerkverbindung unterbrochen.
WM_CTI_NETWO RKOPEN	WM_USER + 22	Netzwerkverbindung steht.
WM_CTI_RAISEC LIENT	WM_USER + 33	Client soll nach vorne gebracht werden.
WM_CTI_ACTIV TECLIENT	WM_USER + 33	Veraltet. Benutzen Sie WM_CTI_RAISECLIENT .
WM_CTI_NETWO RK_DISPATCH	WM_USER + 21	Empfang von CTI-Nachrichten

Siehe auch

[Übersicht C++ CTI API](#) | [CTI_NStartClientCALLBACK](#) | [CTI_NStartClientHWND](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.6.1 WM_CTI_MENUCOMMAND

Menünachrichten werden empfangen.

Nachricht WM_CTI_MENUCOMMAND (WM_USER + 31)**WPARAM**

Im niederwertigen Word ist die Befehls-ID enthalten.

LPARAM

Wird nicht verwendet.

Bemerkung

Diese Nachricht wird gesendet wenn der Benutzer auf das Menü des Verbindungsicons klickt.

Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Events](#) | [CTIClient_SHOWICON](#) | [CTIClient_SHOWMENU](#) |
[CTI_OConnectionIconOptions](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.6.2 WM_CTI_SHOWPHONELIST

Die Anruferliste soll eingeblendet werden.

Nachricht WM_CTI_SHOWPHONELIST (WM_USER + 32)

WPARAM

Wird nicht verwendet.

LPARAM

Wird nicht verwendet.

Bemerkung

Diese Nachricht kommt wenn ein Doppelklick auf das Telefonsymbol in der Taskleiste erfolgt. Ist [CTIClient_PHONEICON](#) aktiviert, so wird dieses Icon bei eingehenden Anrufen eingeblendet. Die Anwendung sollte dann die Anruferliste einblenden sofern vorhanden.



Beispiel

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.6.3 WM_CTI_NETWORKCLOSE

Teilt dem Client mit dass die Netzwerkverbindung unterbrochen wurde.

Nachricht WM_CTI_NETWORKCLOSE (WM_USER + 23)**WPARAM**

Wird nicht verwendet.

LPARAM

Wird nicht verwendet.

Bemerkung

Die Netzwerkverbindung wurde unterbrochen. Ist [CTICLIENT_AUTOCONNECT](#) eingeschaltet, so versucht die Schnittstelle in regelmäßigen Abständen eine neue Kontaktaufnahme.

Beispiel

[SQL-Demo](#)

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.6.4 WM_CTI_NETWORKOPEN

Teilt dem Client mit dass die Netzwerkverbindung wieder steht.

Nachricht WM_CTI_NETWORKOPEN (WM_USER + 22)

WPARAM

Wird nicht verwendet.

LPARAM

Wird nicht verwendet.

Bemerkung

Die Netzwerkverbindung steht. Ab hier kann [CTI_NLogin](#) aufgerufen werden, wenn die Option [CTIClient_AUTOLOGIN](#) geschaltet ist dann ist dies nicht erforderlich.

Beispiel

[SQL-Demo](#)

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.6.5 WM_CTI_RAISECLIENT

Teilt dem Client mit dass er sich in den Vordergrund stellen soll.

Nachricht WM_CTI_RAISECLIENT (WM_USER + 33)**WPARAM**

Wird nicht verwendet.

LPARAM

Wird nicht verwendet.

Bemerkung

Der Befehl wird in der Regel gesendet, wenn das Netzwerksymbol doppelgeklickt wurde. Die Anwendung sollte dann nach vorne gebracht werden.

Siehe auch

[Übersicht Events](#) | [CTIClient_SHOWICON](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.6.6 WM_CTI_NETWORK_DISPATCH

CTI-Nachrichten werden gesendet.

Nachricht WM_CTI_NETWORK_DISPATCH (WM_USER + 21)

WPARAM

Enthält die Größe der gesendeten Struktur.

LPARAM

Zeiger auf eine [CTI_COMMAND_MSG](#) oder [CTI_PHONE_DATA](#) Struktur.

Bemerkung

Fast alle CTI-Events kommen in diesen Strukturen an. Die Interpretation ist abhängig vom Parameter m_dwCommand, der in beiden Strukturen an der gleichen Stelle liegt. Der LPARAM sollte zuerst als LP_CTI_COMMAND_MSG gecastet werden, um m_dwCommand interpretieren zu können. Mögliche Werte für m_dwCommand sind:

Parameter	Wert	Beschreibung	Interpretation von LPARAM
CTI_LOGIN	0x00011001	Erfolg des Einloggens	LP_CTI_COMMAND_MSG
CTI_CALLNUMBER	0x00012001	Rufnummer wird übermittelt	LP_CTI_COMMAND_MSG
CTI_FORWARDINGSTATE	0x00012004	Status der Rufumleitung wird mitgeteilt.	LP_CTI_COMMAND_MSG
CTI_GETREDIRECTION	0x00012004	Veraltet. Benutzen Sie CTI_FORWARDINGSTATE .	LP_CTI_COMMAND_MSG
CTI_OBJECTOFREDIRECT	0x00012006	Rufumleitungsziel informieren	LP_CTI_COMMAND_MSG
CTI_TELLREDIRECTION	0x00012006	Veraltet. Benutzen Sie CTI_OBJECTOFREDIRECT .	LP_CTI_COMMAND_MSG
CTI_PROTOCOLDATA	0x0001200B	Gesprächsdatensatz übermitteln	LP_CTI_PHONE_DATA
CTI_CALLREMINDER	0x0001200C	Anruferinnerung wird geschickt	LP_CTI_PHONE_DATA
CTI_GETRECALL	0x0001200C	Obsoleter. Use CTI_CALLREMINDER .	LP_CTI_PHONE_DATA
CTI_IDENTITY	0x0001200D	Rufnummernidentifikation	LP_CTI_COMMAND_MSG
CTI_GETIDENTITY	0x0001200D	Veraltet. Benutzen Sie CTI_IDENTITY .	LP_CTI_COMMAND_MSG
CTI_LICENSEINFO	0x0001300E	Lizenzinformationen	LP_CTI_COMMAND_MSG
CTI_GETLICENSEDATA	0x0001300E	Veraltet. Benutzen Sie CTI_LICENSEINFO .	LP_CTI_COMMAND_MSG
CTI_SENDUSERSTATE	0x00012011	Status eines Gruppenmitglieds	LP_CTI_COMMAND_MSG

CTI_CONFNOTIFY	0x00012014	Eintritt in eine Konferenz	LP_CTI_COMMAND_MS G
CTI_GETCONF	0x00012014	Veraltet. Benutzen Sie CTI_CONFNOTIFY .	LP_CTI_COMMAND_MS G
CTI_SENDADDRESS	0x00012016	MSN oder Adresse des Anrufs	LP_CTI_COMMAND_MS G
CTI_SERVICEPAUSE	0x00013001	CTI-Server pausiert	LP_CTI_COMMAND_MS G
CTI_CLIENTCLOSE	0x00013002	Clients beenden und neu starten	LP_CTI_COMMAND_MS G
CTI_SERVERMESSAGE	0x00013003	Administrator informiert Clients	LP_CTI_COMMAND_MS G
CTI_INVALIDSTRINGFORMAT	0x00013004	Unterschiedliche Zeichenbreite	LP_CTI_COMMAND_MS G
CTI_USERTOUSER	0x00013005	Nachrichten zwischen Benutzern	LP_CTI_COMMAND_MS G
CTI_DATABASEUSER	0x00013007	Nachrichten von Serverschnittstelle	LP_CTI_COMMAND_MS G
CTI_GROUPMEMBER	0x00013008	Auflistung der Gruppen	LP_CTI_COMMAND_MS G
CTI_CHATSTART	0x0001300A	Chatsitzung beginnen	LP_CTI_COMMAND_MS G
CTI_CHATSTOP	0x0001300B	Chatsitzung beenden	LP_CTI_COMMAND_MS G
CTI_CHATMESSAGE	0x0001300C	Chatnachricht übertragen	LP_CTI_COMMAND_MS G
CTI_LOCALEINFO	0x0001300D	Parameter der Telefonanlage	LP_CTI_COMMAND_MS G
CTI_GETLOCALE	0x0001300D	Veraltet. Benutzen Sie CTI_LOCALEINFO .	LP_CTI_COMMAND_MS G
CTI_CALLSTATE	0x00030002	Status eigener Anrufe	LP_CTI_COMMAND_MS G
SQL_LOGIN	0x00040001	Erfolg des SQL-Einloggens	LP_CTI_COMMAND_MS G
SQL_DATA	0x00040002	Datensätze kommen	LP_CTI_COMMAND_MS G
SQL_END	0x00040004	Abfrage wurde beendet	LP_CTI_COMMAND_MS G
SQL_DBBOOK	0x00040006	Rufnummern aus der Datenbank	LP_CTI_COMMAND_MS G

Beispiel[SQL-Demo](#)[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht Events](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Eine Rufnummer wird übertragen.

Nachricht CTI_CALLNUMBER (0x00012001)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Ist das niederwertige Word TRUE, so handelt es sich um ein Primärgespräch. Ist es FALSE, so gehört die Rufnummer zu einem anklopfenden Anruf. Im höherwertigen Word ist die Rufnummer gespeichert. Ist es TRUE so war der Anruf eingehend.

m_lParam

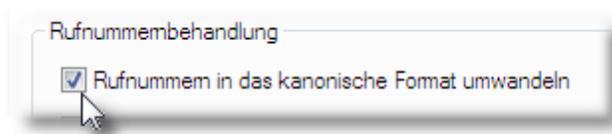
Die Anruf-ID ist hier gespeichert.

m_szMessage

Rufnummer des anderen Teilnehmers.

Bemerkungen

Die Nachricht wird nur geschickt wenn tatsächlich Rufnummern übermittelt wurden. Bei einem ausgehenden Gespräch sollte das immer der Fall sein. Es kann passieren dass diese Nachricht die erste Information zu einem Anruf ist, d. h. es wurden noch keine Nachrichten vom Typ [CTI_CALLSTATE](#) vorher geschickt. Die Rufnummer kann sich während des Gesprächs ändern, wenn z. B. jemand dem Benutzer einen Anruf weitervermittelt. Das Format der Rufnummer kann auf dem Server als kanonisch festgelegt werden (Seite Anlageneinstellungen).

**Beispiel**

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Eine Anruferinnerung wurde dem Benutzer zugesendet.

Nachricht CTI_CALLREMINDER (0x0001200C)**Nachrichtentyp**

CTI_PHONE_DATA

Benutzte Werte*m_unDest*

Durchwahl des Benutzers, der die Anruferinnerung gesendet hat.

m_szNumber

Nummer, welche zurückgerufen werden soll.

*m_szMessage*Zusätzliche Nachricht zum Problem. Hier könnte der Absender erklären, warum *m_szNumber* zurückgerufen werden soll.**Bemerkungen**

Die Benutzer können sich gegenseitig Anruferinnerungen schicken. Somit muss man nicht mehr jedem mitteilen, wer dem zurückrufen soll, der Vorgang wird wesentlich kürzer gestaltet. Anruferinnerungen werden mit [CTI_TSendCallReminder](#) verschickt.

Siehe auch[Übersicht WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Übermittelt den Zustand eigener Gespräche.

Nachricht CTI_CALLSTATE (0x00030002)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte*m_wParam*[Gesprächszustand](#), in dem sich der Anruf befindet.*m_lParam*

Anruf-ID des Gesprächs

Bemerkungen

Der Zustand eines Gesprächs hat sich geändert. Diese Nachricht wird für die Gespräche des jeweiligen Benutzers übertragen, der das Gespräch führt. Sie sollten eine Liste mit aktiven Anrufen im Programm mitführen. Prüfen Sie bei jedem Aufruf von **CTI_CALLSTATE** ob die Anruf-ID in der Liste enthalten ist und fügen Sie diese bei Bedarf ein. Sollte ein Status [LINECALLSTATE_IDLE](#) gesendet werden, so entfernen Sie den Anruf aus der Liste. In der Liste werden sich meist nur ein oder zwei Anrufe befinden.

Zustände der Gruppenmitglieder werden mit dem Event [CTI_SENDUSERSTATE](#) gesendet.

Beispiel[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#) | [Gesprächszustände](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Chatsitzung soll beendet werden.

Nachricht CTI_CHATSTOP (0x0001300B)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

Keine

Bemerkungen

Die Nachricht wird an beide Chatpartner gesendet, wenn einer von ihnen die Chatsitzung schließt. Die Chatfenster sollten daraufhin geschlossen werden.

Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#) | [CTI_EChatEnd](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Eine Chatnachricht wird vom Chatpartner gesendet.

Nachricht CTI_CHATMESSAGE (0x0001300C)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_szMessage

Nachricht, maximal 255 Zeichen

Bemerkungen

Ein Chat mit einem anderen Teilnehmer besteht und dieser hat uns eine Nachricht geschickt. Zeigen Sie diese Nachricht im Ausgabefenster für diese Nachrichten an. Die Nachrichten werden jedes mal komplett übertragen, da der Chatpartner auch markierten Text in seiner Sendebox löschen kann.

Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#) | [CTI_EChatMessage](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Eine Chatnachricht wird vom Chatpartner gesendet.

Nachricht CTI_CHATSTART (0x0001300A)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Ist TRUE, wenn der Aufbau des Chats erfolgreich war.

m_unNumber

Durchwahl des anderen Chatpartners

Bemerkungen

Es Gibt zwei Möglichkeiten, dieses Event zu erhalten:

1. Sie haben mit [CTI_EChatStart](#) um eine Chatsitzung mit einem Teilnehmer gebeten. Diese kann von den Rechten her nicht erlaubt sein oder der Chatpartner hat bereits eine Sitzung. Dann ist *bStartOK* FALSE. Ansonsten bekommen Sie TRUE zurück, in dem Fall können Sie geeignete Fenster für die Chat Ein- und Ausgabe anzeigen.
2. Jemand möchte eine Chatsitzung mit Ihnen beginnen. Anhand von *bstrUser* sehen Sie wer das ist. Wenn Sie die Chatsitzung ablehnen möchten, so rufen Sie hier [CTI_EChatEnd](#) auf. Ansonsten zeigen Sie die Fenster für Ein- und Ausgabe an.

Siehe auch

[Übersicht WM CTI_NETWORK_DISPATCH](#) | [CTI_EChatStart](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Der Client soll geschlossen und auf den Neustart vorbereitet werden.

Nachricht CTI_CLIENTCLOSE (0x00013002)

Nachrichtentyp

CTI_COMMAND_MSG

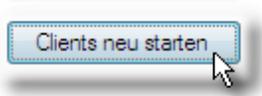
Benutzte Werte

m_IParam

Zeitspanne in Minuten, nach deren Verlauf der Client neu gestartet werden soll, wenn der Wert positiv oder 0.

Bemerkungen

Diese Nachricht schickt der Administrator, wenn er wichtige Dateien austauschen will. Das macht Sinn wenn die Clients aus einem freigegebenen Netzwerkverzeichnis gestartet wurden. Dazu müssen alle Clients heruntergefahren werden. Wenn Sie dieses Feature unterstützen möchten rufen Sie hier [CTI_NPrepareRestart](#) auf und übergeben *m_IParam* als Parameter. Bei einem negativen Wert von *m_IParam* soll die Anwendung nicht neu gestartet werden. Schließen Sie danach Ihre Anwendung. **CTI_CLIENTCLOSE** wird über die Serverkonsole ausgelöst (Seite Benutzerverwaltung).

**Beispiel**

[Mini CTI Client \(C++\)](#)

Siehe auch

[Übersicht WM_CTU_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Konferenzzugehörigkeit hat sich geändert.

Nachricht CTI_CONFNOTIFY (0x00012014)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

TRUE, wenn der Benutzer in eine Konferenz eintritt, FALSE, wenn er sie verlässt.

Bemerkungen

Beim Eintreten in eine und beim Verlassen von Konferenzen wird diese Nachricht gesendet.

Siehe auch

[Übersicht WM_CTU_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Eine benutzerdefinierte Nachricht wurde vom Server geschickt.

Nachricht CTI_DATASETUSER (0x00013007)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_lParam

Typ des benutzerdefinierten Kommandos

m_szMessage

Übermittelte Parameter

Bemerkungen

TAPIMaster® besitzt auf der Serverseite eine zweite Schnittstelle. Normalerweise werden Datenbankabfragen über die ODBC-Schnittstelle des Servers getätigt. Alternativ dazu kann die Serverschnittstelle jedoch in einer Datenbank oder anderen Anwendung eingebunden werden. Über [CTI_EUserSendToDataBase](#) können dann benutzerdefinierte Daten getauscht werden.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#) | [CTI_EUserSendToDataBase](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die eigene Rufumleitung wird übertragen.

Nachricht CTI_FORWARDINGSTATE (0x00012004)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Wenn TRUE, dann ist eine Rufumleitung geschaltet.

m_szMessage

Nummer, zu der umgeleitet wird.

Bemerkungen

Die Einstellungen für die Rufumleitungen werden zentral auf dem CTI-Server gespeichert. Deshalb wird diese Nachricht nach der Anmeldung gesendet. Bei Änderungen an der Rufumleitung wird **CTI_FORWARDINGSTATE** ebenfalls aufgerufen.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Teilt den Clients beim Start die Mitglieder ihrer Gruppe und deren Zustand mit.

Nachricht CTI_GROUPMEMBER (0x00013008)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Bitmuster. Bedeutung der Flags:

0x00000001 Gruppenmitglied ist angemeldet.

0x00000002 Es gibt aktive Gespräche.

0x00000004 Es gibt eingehende Gespräche, die nicht entgegengenommen sind.

m_lParam

Wenn hier das Bit 0x01000000 gesetzt ist, dann hat das Gruppenmitglied eine Rufumleitung geschaltet.

m_unNumber

Durchwahl des Gruppenmitglieds

m_szMessage

Name des Gruppenmitglieds. Hier wird der Anmeldename verwendet. Dieser ist erst verfügbar wenn sich dieser Kollege mindestens einmal eingeloggt hat, da sich diese Daten automatisch einpflegen.

Bemerkungen

Bei jedem Anmelden werden die Daten der Gruppenmitglieder geschickt. Pro Gruppenmitglied wird eine Nachricht gesendet. Wird das Event mit einem leeren String in *m_unNumber* aufgerufen, so ist die Übertragung beendet. Mit diesen Events sind gleich zu Programmstart die Gesprächszustände der Gruppenmitglieder bekannt. Diese werden zur Laufzeit mit [CTI_SENDUSERSTATE](#) aktualisiert.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Rufnummernidentifikation kommt an.

Nachricht CTI_IDENTITY (0x0001200D)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_IParam

Anruf-ID des Gesprächs

m_szMessage

Name des Anrufers, bis zu 255 Zeichen

Bemerkungen

Auf dem Server wurde eine Rufnummernidentifikation durchgeführt. Diese kann von der ODBC-Schnittstelle oder der Server Programmierschnittstelle kommen. Neben dem Namen können in *m_szMessage* noch weitere Daten übertragen werden. Die Nachricht wird auch gesendet wenn auf dem Client eine Anruferidentifikation über Telefonbuch-CD's stattfindet.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Server und Client passen von der Plattform her nicht zueinander.

Nachricht CTI_INVALIDSTRINGFORMAT (0x00013004)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

Keine

Bemerkungen

Es gibt eine TAPIMaster® Version für ANSI-Standard und eine für Unicode. Letztere arbeitet intern mit 2Byte Zeichen. Client und Server müssen immer der gleichen Version angehören. In Deutschland wurde bisher nur die ANSI-Version vertrieben, so dass der Fehler nicht auftreten sollte.

Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Liefert erweiterte Lizenzinformationen.

Nachricht CTI_GETLICENSEDATA (0x0001300E)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_szMessage

Erweiterte Lizenzinformationen. Die übermittelten Werte sind von geschweiften Klammern eingerahmt und durch Semikolons getrennt. Formatbeispiel:

{License for: Musterfirma;Street: Musterstraße 1;City: 12345 Königs-Musterhausen;Phone: +49 (5555) 664488;Web: http://www.musterfirma.de}

Bemerkungen

Die Nachricht wird nach dem Anmelden gesendet.

Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Liefert Anlagenparameter und Landeseinstellungen.

Nachricht CTI_GETLOCALE (0x0001300D)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Im niederwertigen Word ist die Länge der internen Nummern enthalten, im höherwertigen Word die Landeskennung.

m_lParam

Anlagenrufnummer. Wird bei Telefonanlagen mit Amtsanschluss verwendet. Bei einer Rufnummer +49 (69) 123456-789 wäre 123456 die Anlagenrufnummer.

m_szMessage
Vorwahl ohne führende 0

Bemerkungen

Die Funktion liefert bei jeder Anmeldung diese lokalen Parameter zurück, die dann im Programm weiter verwendet werden können.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Anmeldung am CTI-Server wird bestätigt.

Nachricht CTI_LOGIN (0x00011001)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Erfolgsstatus der Anmeldung, ist TRUE wenn der Benutzer sich anmelden konnte.

m_lParam

Ist TRUE wenn es sich um eine Lizenzversion handelt.

Bemerkungen

Eine erfolgreiche Anmeldung ist die Voraussetzung dafür dass die meisten Befehle abgesetzt werden können und Events gesendet werden.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Auf den Benutzer wird eine Rufumleitung geschaltet.

Nachricht CTI_OBJECTOFREDIRECT (0x00012006)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Auf den Teilnehmer wurde eine Rufumleitung geschaltet, wenn *m_wParam* TRUE ist. Wenn FALSE wurde die Rufumleitung wieder abgeschaltet.

m_szNumber

Durchwahl des Teilnehmers, der zu diesem Anschluß umleitet.

Bemerkungen

Wenn ein anderer Teilnehmer eine Rufumleitung hierher schaltet wird dieser Event aufgerufen. Ebenso wenn dieser die Rufumleitung wieder herausnimmt. Sollten Rufumleitungen auf diesen Anschluß geschaltet sein so wird die Meldung auch nach Anmeldung übertragen.

Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#) | [CTI TSetForwarding](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Gesprächsdatensätze werden übermittelt.

Nachricht CTI_PROTOCOLDATA (0x0001200B)**Nachrichtentyp**

CTI_PHONE_DATA

Benutzte Werte

m_dwStart

Gesprächsbeginn, Format time_t

m_dwEnd

Gesprächsende, Format time_t

m_dwFlags

Bitflag für Erfolg und Richtung des Gesprächs

0x00000002 Der Anruf war erfolgreich.

0x00000004 Es war ein eingehender Anruf.

m_unActor

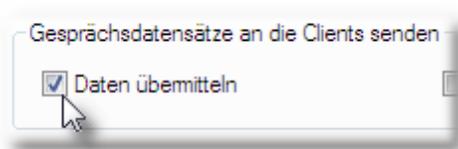
Eigene Durchwahl

m_szNumber

Rufnummer des Gesprächspartners

Bemerkungen

Nach Ende jedes Gesprächs wird ein solcher Gesprächsdatensatz übertragen. Ist der Benutzer abgemeldet, so werden die Daten auf dem CTI-Server gepuffert und bei der nächsten Anmeldung gesendet. Die Daten werden nur gesendet, wenn auf dem Server diese Option eingeschaltet ist. Auf der Seite "Datenbankschnittstelle" der Serverkonsole kann die Übermittlung zugeschaltet werden.

**Siehe auch**

[Übersicht WM CTI NETWORK DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Adresse, unter welcher der Anruf einging wird gesendet.

Nachricht CTI_SENDADDRESS (0x00012016)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Wenn TRUE dann wird die Adresse des Telefons in *m_szMessage* übermittelt, bei FALSE ist es die Rufnummer.

m_lParam

Anruf-ID des Gesprächs

m_szMessage

Adresse welche angerufen wurde oder Rufnummer.

Bemerkungen

Diese Nachricht wird bei eingehenden Anrufen gesendet. Ein ISDN-Telefon kann auf mehrere MSNs reagieren. Nehmen wir z. B. einen Heimarbeitsplatz. Hier wird es eine MSN für Privatgespräche und eine MSN für dienstliche Anrufe geben. Am Monatsende soll aber ausgewertet werden, wieviel Zeit dienstlich telefoniert wurde. Über die angerufene MSN in *m_szMessage* können die Anrufe zugeordnet werden.

Eine andere Möglichkeit trifft man bei Telefonanlagen. Eine Leitung kann mehrere interne Durchwahlen haben. Die 210 kann für normale und interne Gespräche verwendet werden, die 310 ist mit einer Hotline verbunden. In *m_szMessage* würde entweder die 210 oder die 310 mitgeliefert.

TAPIMaster® versucht, eine der beiden Angaben zu ermitteln und zu senden. Adressen werden jedoch von nur von wenigen Treibern unterstützt. Was gesendet wird, ist von der Anlage bzw. dem Telefon und den Treibern abhängig. Es ist auch möglich, dass diese Nachricht nicht gesendet wird oder mehrmals während eines Anrufs kommt.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Der Status eines Gruppenmitglieds hat sich geändert.

Nachricht CTI_SENDUSERSTATE (0x00012011)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Bitmuster. Bedeutung der Flags:

0x00000001 Gruppenmitglied ist angemeldet.

0x00000002 Es gibt aktive Gespräche.

0x00000004 Es gibt eingehende Gespräche, die nicht entgegengenommen sind.

m_IParam

Wenn hier das Bit 0x01000000 gesetzt ist, dann hat das Gruppenmitglied eine Rufumleitung geschaltet.

m_unNumber

Durchwahl des Gruppenmitglieds

Bemerkungen

Die Nachricht wird übertragen, wenn sich der Zustand eines Gruppenmitglieds geändert hat. Der Benutzer hat somit die Möglichkeit zu reagieren wenn ein Gruppenmitglied ein Gespräch nicht annimmt oder ein anklopfendes Gespräch erhält. Seine Erreichbarkeit kann ebenfalls immer eingesehen werden.

Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#), [CTI GROUPEMEMBER](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Der Administrator schickt dem Benutzer eine Nachricht.

Nachricht CTI_SERVERMESSAGE (0x00013003)

Nachrichtentyp

CTI_COMMAND_MSG

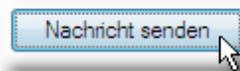
Benutzte Werte

m_szMessage

Nachricht, maximal 255 Zeichen

Bemerkungen

Der Administrator schickt eine Nachricht an den Benutzer, welche in *m_szMessage* enthalten ist. Die Nachrichten werden von der Serverkonsole auf der Seite "Benutzerverwaltung" gesendet.



Siehe auch

[Übersicht WM CTI NETWORK DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Der CTI-Server wurde angehalten.

Nachricht CTI_SERVICEPAUSE (0x00013001)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte*m_wParam*

Wenn TRUE, so arbeitet der Server, bei FALSE wurde er angehalten.

Bemerkungen

Der CTI-Server kann zu Wartungszwecken angehalten werden. Sie sollten dann im Programm bestimmte Schalter wie Wählknöpfe deaktivieren. Wenn der CTI-Server seinen Betrieb wieder aufnimmt können diese aktiviert werden.

Siehe auch[Übersicht WM CTI NETWORK DISPATCH](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Ein anderer Benutzer schickt eine Nachricht.

Nachricht CTI_USERTOUSER (0x00013005)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte*m_unNumber*

Durchwahl des anderen Teilnehmers

m_szMessage

Textnachricht, die er verschickt, maximal 255 Zeichen.

Bemerkungen

Die Benutzer können sich untereinander mit [CTI_ESendUserToUser](#) Nachrichten zukommen lassen. Wenn Sie diese Funktion implementieren können Sie möglicherweise an manchen Arbeitsplätzen auf einen Mailclient verzichten.

Siehe auch[Übersicht WM CTI NETWORK DISPATCH](#) | [CTI_ESendUserToUser](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

SQL-Abfragen werden beantwortet.

Nachricht SQL_BEGIN (0x00040002)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte*m_wParam*

Nummer der Anfrage, welche mit [CTI_SQLBegin](#) gestartet wurde.

m_IParam

Anzahl der zurückgelieferten Datensätze, ist 0 wenn keine Daten vorhanden waren und -1 wenn die Abfrage fehlschlug.

m_szMessage

Text mit den Abfrageergebnissen

Bemerkungen

Dieser Event wird immer dann aufgerufen wenn Daten mit [CTI_SQLBegin](#) oder [CTI_SQLNext](#) angefordert wurden. Die Daten können im XML-Format oder als Text kommen. Die Abfrage kann fehlschlagen wenn eine Abfrage bereits terminiert wurde oder bereits alle Daten gelesen wurden.

Beispiel

[SQL-Demo](#)

[Datenbankabfragen](#)

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#) | [CTI_SQLBegin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Datensätze aus der Rufnummernsuche werden übermittelt.

Nachricht SQL_DBBOOK (0x00040006)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Anzahl der übermittelten Datensätze

m_szMessage

Text mit den Datensätzen

Bemerkungen

Der Benutzer hat eine Anfrage an die Datenbank vom Typ [CTI_SQLGetDbBook](#) gestartet. Die Datensätze stehen in geschweiften Klammern. Name, Rufnummer und Zusatzinformation sind durch Semikolon getrennt. Die Zusatzinformationen müssen nicht zwangsläufig vorhanden sein. Umfang und Format der ankommenden Daten werden in den Datenbankeinstellungen des CTI-Servers festgelegt.

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#) | [CTI_SQLGetDbBook](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Eine SQL-Abfrage wurde beendet.

Nachricht SQL_END (0x00040004)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Nummer der Datenbankabfrage, welche bei [CTI_SQLBegin](#) vergeben wurde.

m_lParam

Ist TRUE wenn die Abfrage beendet werden konnte.

Bemerkungen

Der Event wird auf den Befehl [CTI_SQLEnd](#) hin gesendet. Werden mit einem [CTI_SQLBegin](#) oder [CTI_SQLNext](#) alle Datensätze gelesen, so kommt dieser Event selbständig. Wird eine Datenbankabfrage nicht terminiert, so terminiert der Server diese automatisch nach einigen Minuten und sendet diesen Event.

Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#) | [CTI_SQLEnd](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Anmeldung an die Datenbank wird bestätigt.

Nachricht SQL_LOGIN (0x00040001)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Ist TRUE, wenn die Anmeldung erfolgreich war.

Bemerkungen

Bevor ein Anwender die Datenbank benutzt muss er sich dort mit [CTI_SQLLogin](#) anmelden. Der Server schickt daraufhin diese Nachricht zurück. Wenn bei der Anmeldung ein falsches Paßwort übermittelt wurde oder die Datenbankverbindung nicht aktiv ist wird die Anmeldung abgewiesen. Die Datenbankverbindung wird auf der Serverkonsole auf der Seite "Datenbankschnittstelle" zugeschaltet.



Beispiel

[SQL-Demo](#)

Siehe auch

[Übersicht WM_CTI_NETWORK_DISPATCH](#) | [CTI_SQLLogin](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.7 Strukturen

Die [Callback-Funktionen](#) senden in ihren Rückgabewerten Zeiger auf Strukturen, in denen dann die Daten enthalten sind. Datenbankabfragen können XML-Strukturen zurück liefern.

[CTI_COMMAND_MSG](#)
[CTI_PHONE_DATA](#)
[Datenbankabfragen](#)

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.7.1 CTI_COMMAND_MSG

Definition

```
#define CTI_NUMBER_SIZE      8
#define CTI_MESSAGE_SIZE    256

typedef struct cti_command_tag_msg
{
    DWORD m_dwSize;
    DWORD m_dwCommand;
    WPARAM m_wParam;
    LPARAM m_lParam;
    union tag_number
    {
        CTI_NUMBER wNumber;
        TCHAR szNumber[CTI_NUMBER_SIZE];
    }m_unNumber;
    TCHAR m_szMessage[CTI_MESSAGE_SIZE];
}CTI_COMMAND_MSG, FAR* LP_CTI_COMMAND_MSG;
```

Bemerkung

In diesen Strukturen werden die meisten Daten über das Netz geschickt. Die Größe der Struktur ist in `m_dwSize` enthalten, diese richtet sich hauptsächlich nach der Länge der verschickten Nachricht plus der vorhergehenden Daten. In `m_dwCommand` ist der [Befehlsparameter](#) enthalten. Die übrigen Werte werden je nach Befehl interpretiert. Insbesondere bei SQL-Nachrichten wie [SQL_DATA](#) kann der empfangene Text deutlich größer als 256 Zeichen sein.

Siehe auch

[Strukturen](#) | [WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.7.2 CTI_PHONE_DATA

Definition

```
#define CTI_NUMBER_SIZE      8
#define PHONE_NUMBER_SIZE   32
#define CTI_MESSAGE_SIZE    256

typedef struct cti_phone_data_tag
{
    DWORD m_dwSize;
    DWORD m_dwCommand;
    time_t m_dwStart;
    time_t m_dwEnd;
    DWORD m_dwFlags;
    union tag_Actor
    {
        CTI_NUMBER wNumber;
        TCHAR szNumber[CTI_NUMBER_SIZE];
    }m_unActor;
    TCHAR m_szNumber[PHONE_NUMBER_SIZE];
    union tag_Dest
    {
        CTI_NUMBER wNumber;
        TCHAR szNumber[CTI_NUMBER_SIZE];
    }m_unDest;
    TCHAR m_szMessage[CTI_MESSAGE_SIZE];
}CTI_PHONE_DATA, FAR* LP_CTI_PHONE_DATA;
```

Bemerkung

In dieser Struktur werden [Gesprächsdatensätze](#) und [Anruferinnerungen](#) übermittelt.

Siehe auch

[Strukturen](#) | [WM_CTI_NETWORK_DISPATCH](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.7.3 Datenbankabfragen

Einstellung des Rückgabetyps

Die TAPIMaster®-Datenbankschnittstelle kann entweder im Textformat oder als XML kommen. Das Format kann in der Serverkonsole auf der Seite "Datenbankschnittstelle" eingestellt werden.

**XML-Format**

Hier ein Beispiel für eine Abfrage an die Beispieldatenbank, welche sich im Ordner "Datenbank" des Programmierpaketes befindet. Die Abfrage lautet:

```
SQLBegin(„SELECT * FROM Members“,2)
```

Die Antwort:

```
<?xml version="1.0" standalone="yes"?>
<DATAPACKET Version="2.0">
  <METADATA>
    <FIELDS>
      <FIELD tagname="MemberNumber" fieldtype="i4" />
      <FIELD tagname="GivenName" fieldtype="string" WIDTH="40" />
      <FIELD tagname="Surname" fieldtype="string" WIDTH="32" />
      <FIELD tagname="DayOfBirth" fieldtype="datetime" WIDTH="19" />
      <FIELD tagname="Tax" fieldtype="numeric" WIDTH="19" />
      <FIELD tagname="VoteRights" fieldtype="i1" WIDTH="3" />
      <FIELD tagname="DrivingLicense" fieldtype="boolean" />
      <FIELD tagname="ThrowWidth" fieldtype="r8" WIDTH="53" />
    </FIELDS>
  </METADATA>
  <ROWDATA>
    <ROW>
      <MemberNumber>1</MemberNumber>
      <GivenName>Walter</GivenName>
      <Surname>Miller</Surname>
      <DayOfBirth>1962-11-14 00:00:00</DayOfBirth>
      <Tax>15.5000</Tax>
      <VoteRights>20</VoteRights>
      <DrivingLicense>1</DrivingLicense>
      <ThrowWidth>12.5</ThrowWidth>
    </ROW>
    <ROW>
      <MemberNumber>2</MemberNumber>
      <GivenName>Paul</GivenName>
      <Surname>Smith</Surname>
      <DayOfBirth>1959-12-26 00:00:00</DayOfBirth>
      <Tax>12.0000</Tax>
      <VoteRights>10</VoteRights>
      <DrivingLicense>1</DrivingLicense>
      <ThrowWidth>11.99</ThrowWidth>
    </ROW>
  </ROWDATA>
</DATAPACKET>
```

Textformat

Als normaler Text haben die gleichen Daten wie oben folgendes Aussehen:

```
{HEADER ColCount=8}
{COLUMNS (Type=4 Text=MemberNumber);(Type=12 Text=GivenName);(Type=12 Text=Surname);
(Type=93 Text=DayOfBirth);(Type=2 Text=Tax);(Type=-6 Text=VoteRights);(Type=-7
Text=DrivingLicense);(Type=8 Text=ThrowWidth);}
{(Value0=1);(Value1=Walter);(Value2=Miller);(Value3=1962-11-14 00:00:00);
(Value4=15.5000);(Value5=20);(Value6=1);(Value7=12.5);}
{(Value0=2);(Value1=Klaus);(Value2=Smith);(Value3=1959-12-26 00:00:00);
(Value4=12.0000);(Value5=10);(Value6=1);(Value7=11.99);}
```

SQL-Datentypen

Die von der TAPIMaster®-Schnittstelle unterstützten Typen sind hier aufgelistet:

Typ	Wert
char	1
numeric	2
decimal	3
i4	4
i2	5
float	6
r4	7
r8	8
date	9
time	10
timestamp	11
string	12
datetime	93
string	-1
i8	-5
i1	-6
boolean	-7

Siehe auch

[Strukturen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.8 Konstanten

Verwendung der Konstanten

In der TAPIMaster® API werden einige Konstanten verwendet. Teilweise kommen diese von der TAPI-Schnittstelle, z. B. die [Gesprächszustände](#). Daneben gibt es noch diverse Flags.

[Gesprächszustände](#)

[Verbindungsoptionen](#)

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.8.1 Gesprächszustände

Ein Anruf befindet sich in einem bestimmten Zustand, je nachdem das Gespräch erst aufgebaut wird, aktiv ist oder gerade beendet wird. Mit dem Zustand IDLE wird das Gespräch beendet und kann aus Ihrer internen Gesprächsverwaltung entfernt werden. Sie finden diese Gesprächszustände auch in der Datei TAPI.H in Ihrer Entwicklungsumgebung. Dort sind diese mit dem Präfix LINECALLSTATE_ versehen.

Bezeichnung	Wert (hexadezi mal)	Beschreibung
IDLE	0x0001	Anlage bereit
OFFERING	0x0002	Eingehender Anruf
ACCEPTED	0x0004	Der Teilnehmer / das Gerät behandelt den Anruf
DIALTONE	0x0008	Hörer abgenommen
DIALING	0x0010	Anrufer wählt
RINGBACK	0x0020	Es klingelt an der Gegenstelle
BUSY	0x0040	Das Telefon ist beschäftigt
SPECIALINFO	0x0080	Herstellerspezifische Info
CONNECTED	0x0100	Teilnehmer sind verbunden
PROCEEDING	0x0200	Die Nummer ist komplett, der Anruf wird aufgebaut
ONHOLD	0x0400	Gespräch wird gehalten
CONFERENCE D	0x0800	Gespräch in Konferenz
ONHOLDPEND CONF	0x1000	Konferenz ist eingeleitet worden
ONHOLDPENDT RANSFER	0x2000	Gespräch wartet auf Weitervermittlung
DISCONNECTE D	0x4000	Gespräch beendet
UNKNOWN	0x8000	Unbekannter Status

Siehe auch

[Übersicht Konstanten](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.3.8.2 Verbindungsoptionen

Verwendung der Verbindungsoptionen

Vor Verbindungsaufnahme werden diese Optionen mit Hilfe der Funktion [CTI_OSetOptions](#) gesetzt. Diese Konstanten sind in der Datei CTITYPES.H definiert.

Konstante	Wert (hexadezimal)	Beschreibung
CTCLIENT_AUTOCONNECT	0x00000002	Selbständig Verbindung halten
CTCLIENT_SHOWICON	0x00000004	Verbindungsicon
CTCLIENT_CHECKCONNECTION	0x00000008	Verbindungsdaten beim Start prüfen
CTCLIENT_AUTOLOGIN	0x00000010	Selbständig anmelden
CTCLIENT_SHOWMENU	0x00000020	Tray Icon hat Menü
CTCLIENT_HOTKEY	0x00000040	Schnellwahl über Tastenkürzel
CTCLIENT_PHONEICON	0x00000080	Anrufsymbol
CTCLIENT_ASSISTEDTELEPHONY	0x00000100	Client unterstützt Assisted Telephony
CTCLIENT_AUTOANSWER	0x00000200	Eingehende Anrufe automatisch entgegennehmen

Siehe auch

[Übersicht Konstanten](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Unterstützung von Assisted Telephony zuschalten.

Flag CTCLIENT_ASSISTEDTELEPHONY (0x00000100)**Bemerkungen**

Nimmt Befehle für Assisted Telephony Funktionen entgegen. Wenn aus einem Fremdprogramm die Windows TAPI Funktion `tapiRequestMakeCall` aufgerufen wird, so startet die Schnittstelle einen Anruf.

Voreinstellung

Aktiviert

Siehe auch

[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Gespräche werden automatisch angenommen.

Flag CTIClient_AUTOANSWER (0x0000200)**Bemerkungen**

Eingehende Gespräche werden automatisch angenommen. Diese Option macht Sinn in Callcentern. Das Telefon / Headset muss ins Freisprechen schalten können und der TAPI Treiber dies unterstützen.

Voreinstellung

Deaktiviert

Siehe auch

[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Netzwerkverbindung wird selbständig gehalten.

Flag CTIClient_AUTOCONNECT (0x0000002)**Bemerkungen**

Beim Programmstart stellt die CTI-Schnittstelle selbständig die Verbindung zum Server her. Wird diese Verbindung unterbrochen, so versucht der Client diese wieder aufzubauen.

Voreinstellung

Aktiviert

Siehe auch

[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Selbständig an CTI-Server anmelden.

Flag CTIClient_AUTOLOGIN (0x0000010)**Bemerkungen**

Die Schnittstelle meldet sich nach Aufbau der Netzwerkverbindung selbständig am Server an, [CTI_NLogin](#) muss nicht separat aufgerufen werden.

Voreinstellung

Aktiviert

Siehe auch

[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#) | [CTI_OSetOwnerNumber](#) | [CTI_NLogin](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Die Verbindungsdaten werden beim Start geprüft.

Flag CTIClient_CHECKCONNECTION (0x00000008)**Bemerkungen**

Beim Programmstart prüft die Schnittstelle die Verbindungsparameter auf Vollständigkeit. Das wären Servername, Portnummer und interne Durchwahl. Fehlt ein Parameter, so bringt die Schnittstelle den [Anmeldedialog](#) nach vorne.

Voreinstellung

Aktiviert

Siehe auch

[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Schaltet die Schnellwahl über Tastaturcodes zu.

Flag CTIClient_HOTKEY (0x00000040)**Bemerkungen**

Legt fest, ob markierter Text mit einer Tastenkombination angewählt werden kann. Das kann eine der Tasten F1 bis F12 sein, Zusätzlich können die Umschalt- und / oder die Strg-Taste gedrückt sein. Befindet sich eine wählbare Nummer im Text, so wird diese angewählt. Die Schnellwahl- und Zusatz Tasten werden mit [CTI_OSetDialKey](#) festgelegt.

Voreinstellung

Deaktiviert

Siehe auch

[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#) | [CTIClient_QUICKMOUSEDIAL](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Ein Telefonsymbol wird bei eingehenden Anrufen eingeblendet.

Flag CTIClient_PHONEICON (0x00000080)**Bemerkungen**

Bei eingehenden Anrufen kann in der Taskleiste ein kleines Telefonsymbol eingeblendet werden, analog dem Briefsymbol in MS Outlook. Im Tooltip des Icons wird dann die Rufnummer sofern vorhanden angezeigt. Das Icon verfügt über ein kleines Menü, mit dem der Anrufer zurückgerufen

werden kann. Das Icon wird mit [CTI_TRemovePhoneIcon](#) entfernt. Ein Doppelklick auf das Icon sendet die Nachricht [WM_CTL_SHOWPHONELIST](#), hier sollten Sie die Anruferliste sofern vorhanden anzeigen.

**Voreinstellung**

Aktiviert

Siehe auch[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#)

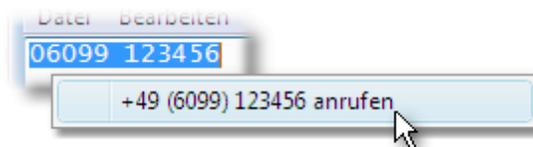
[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Zuschaltung der Schnellwahl über Kontextmenü.

Flag CTIClient_QUICKMOUSEDIAL (0x00000400)**Bemerkungen**

Mit Hilfe dieser Option kann markierter Text über ein Kontextmenü angewählt werden. Befindet sich eine wählbare Nummer im Text, so blendet sich ein solches Kontextmenü ein:

**Voreinstellung**

Deaktiviert

Siehe auch[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#) | [CTIClient_HOTKEY](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Ein Verbindungssymbol wird angezeigt.

Flag CTIClient_SHOWICON (0x00000004)**Bemerkungen**

In der Taskleiste wird ein Verbindungssymbol angezeigt, welches den Status der Netzwerkverbindung erkennen lässt. Es kann mit einem benutzerdefinierten Menü und Tooltip versehen werden. Die Icons für stehende unterbrochene Verbindung können durch eigene Icons ausgetauscht werden.

**Voreinstellung**

Aktiviert

Siehe auch[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#) | [CTI_OConnectionIconOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Das Verbindungsicon wird mit einem Menü versehen.

Flag CTICLIENT_SHOWMENU (0x00000020)**Bemerkungen**

Dem Verbindungsicon in der Taskleiste lässt sich ein Menü zuordnen. Es erscheint dann wenn der Benutzer mit der rechten Maustaste auf das Icon klickt. Mit [CTI_OConnectionIconOptions](#) wird das Menühandle übergeben. Die Events des Menüs werden mit OnMenuCommand geliefert.

[CTICLIENT_SHOWICON](#) muss aktiviert sein.

Voreinstellung

Deaktiviert

Siehe auch[Übersicht Verbindungsoptionen](#) | [CTI_OSetOptions](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.1.4 Shortcut Dateien

Im Installationsverzeichnis des Clients befindet sich eine Datei mit dem Namen FCTISCUT.EXE. Auf der Clientseite werden Links auf dieses Program verwendet, um Kurzwahlkosten auf dem Desktop darzustellen. Das Programm kann aber auch direkt aufgerufen werden:

```
<PATH>/FCTISCUT #-C:Command Arg1 Arg2 ... Argn
```

Statt eines Aufrufs von `tapiRequestMakeCall` wird auf Terminal Servern diese Vorgehensweise empfohlen.

Mögliche Befehle sind:

Anwahl

```
<PATH>/FCTISCUT #-C:MAKECALL 089123456789
```

Der Client ruft die Nummer 089123456789 an.

Auflegen

```
<PATH>/FCTISCUT #-C:DROP
```

Der Client legt auf.

Gespräch annehmen

```
<PATH>/FCTISCUT #-C:ANSWER
```

Der Client nimmt das Gespräch an.

Siehe auch

[Befehlsreferenz Client](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.2 Programmierbeispiele

Sie finden hier Programmierbeispiele für C++, Delphi, VB/Excel. Die Verwendung der C++ und ActiveX Schnittstelle wird gezeigt. Installieren Sie den CTI-Client, bevor Sie mit den Beispielen arbeiten.

[ClientCommands](#)

[Einbindung in VB](#)

[Mini CTI Client \(C++\)](#)

[Mini CTI Client \(Delphi\)](#)

[SQL-Demo](#)

Siehe auch

[Clientschnittstelle](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

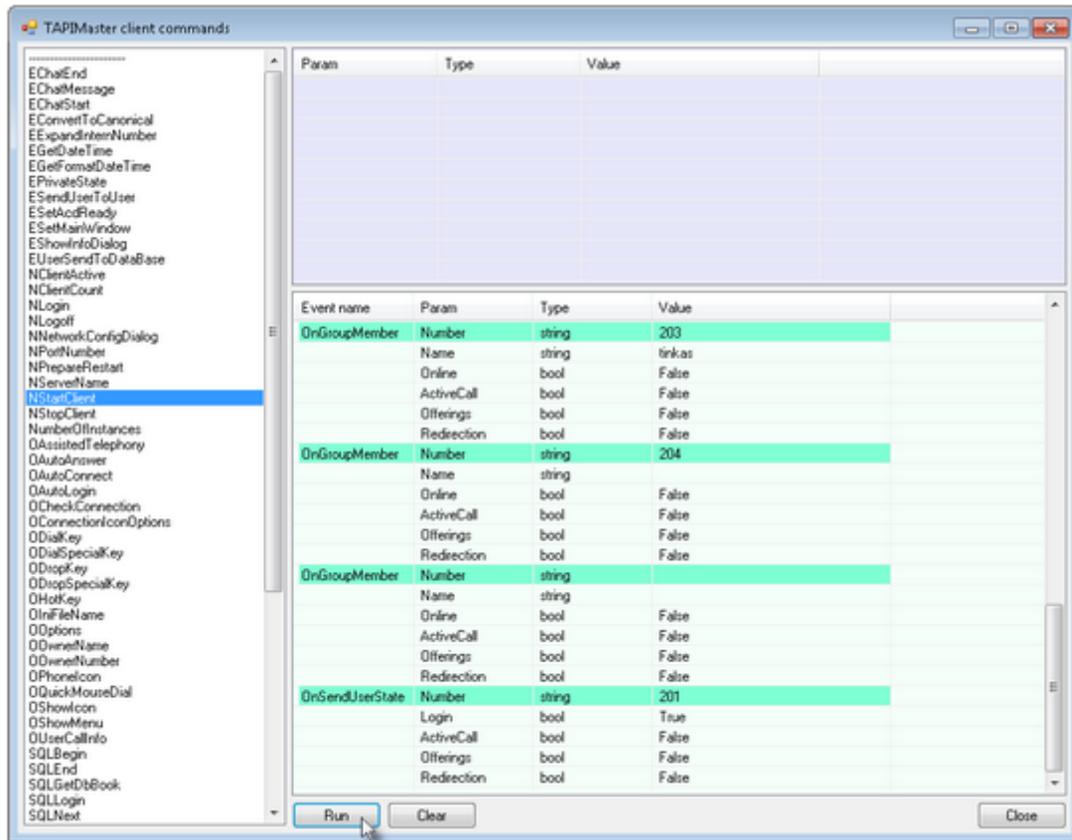
1.2.2.1 ClientCommands

Benötigte Dateien

Einige [Dateien](#) der Client-Installation werden benötigt. Die FCTICLNT.DLL muss registriert sein.

Beispiel

Mit Hilfe des Programms kann jeder Befehl der Client Schnittstelle einzeln aufgerufen werden. Das Programm ist ähnlich dem [CTI Browser](#).



Siehe auch

[Programmierbeispiele](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

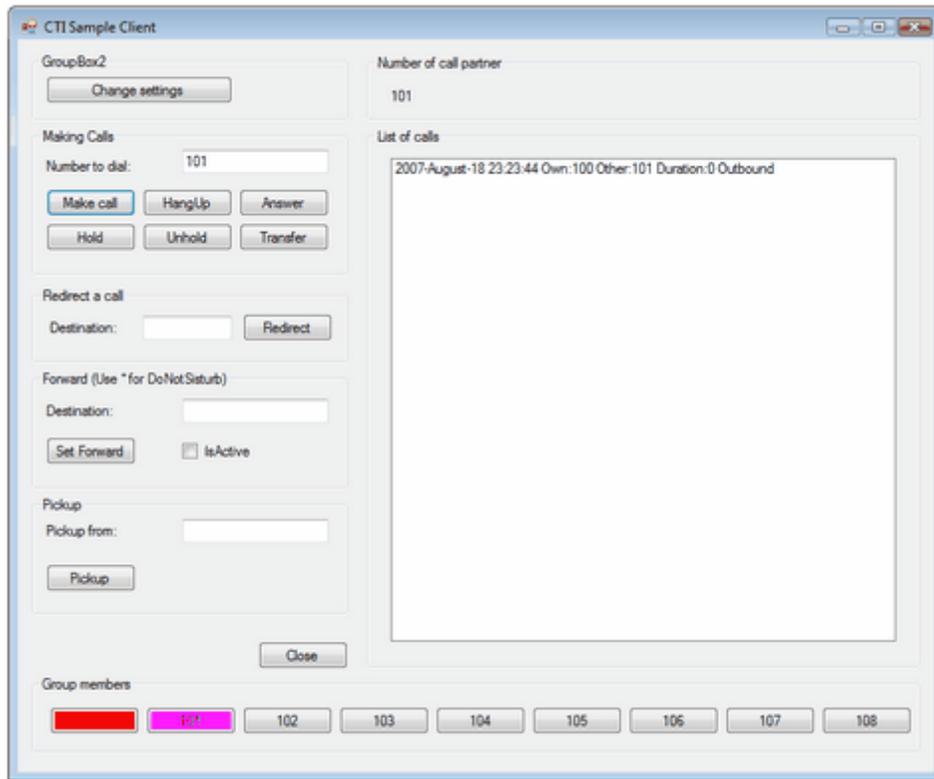
1.2.2.2 Einbindung in VB

Benötigte Dateien

Einige [Dateien](#) der Client-Installation werden benötigt. Die FCTICLNT.DLL muss registriert sein.

Beispiel

Ein VB Client zeigt die wichtigsten Funktionen.



Siehe auch

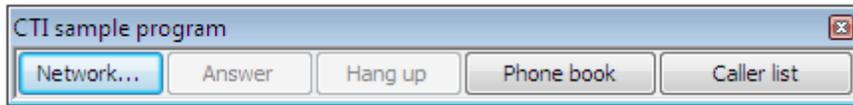
[Programmierbeispiele](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.2.3 Mini CTI Client C++

Dieser Miniclient ist in C++ (MFC, Visual C++ 7.0) geschrieben und enthält die wichtigsten Telefoniefunktionen.



Es gibt ein kleines Telefonbuch, mit der EINGF-Taste können neue Namen eingegeben werden, mit der ENTF-Taste lassen sie sich löschen.

Name	Number
Paula	089-65465456
Walter	069564445646

Die eingehenden Anrufe werden in einer Liste festgehalten. Mit der ENTF-Taste werden sie gelöscht. Das Telefonbuch dient dabei der Anruferidentifikation. Mit einem Doppelklick auf die Liste wird der Anrufer zurückgerufen.

Time	Number	Caller
05.10.2006 21:32:43	102	
05.10.2006 21:32:38	102	
05.10.2006 21:32:29	102	

Siehe auch

[Programmierbeispiele](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.2.4 Mini CTI Client Delphi

Dieser Miniclient ist in Delphi 6 geschrieben und enthält die wichtigsten Telefoniefunktionen. Hier wird die ActiveX Schnittstelle verwendet.



Siehe auch

[Programmierbeispiele](#)

[Send feedback to TAPIMaster@](#)

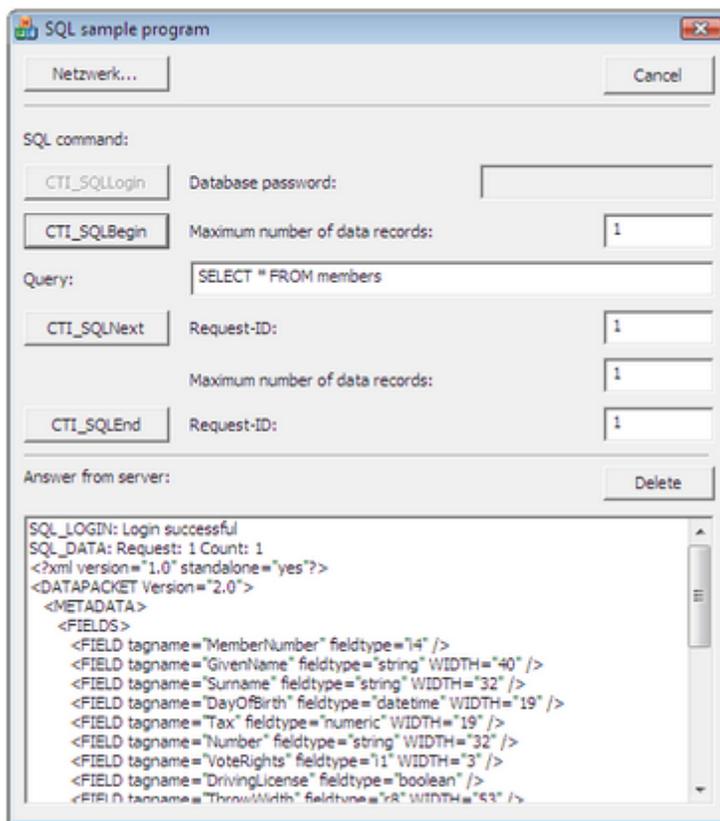
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.2.2.5 SQL-Demo

Die SQL-Demo ist ein C++ Programm, basierend auf der MFC, Visual C++ 7.0. Sie finden es im Ordner CPlusPlus\SQLDemo des CTI-SDK's. Mit dem Beispiel können Sie die SQL-Befehle ausprobieren. Das Beispiel enthält keine Telefoniefunktionen und zeigt nur den Datenbankzugriff. Es ist erforderlich dass auf der Serverkonsole eine Datenbank über ODBC angebunden ist. Der Client sollte auf dem Rechner installiert sein und funktionieren.

SQL-Schnittstelle über ODBC

Machen Sie sich bitte vorher mit den [Befehlen](#) und Events der [SQL-Schnittstelle](#) vertraut. Dann starten Sie das Programm. Geben Sie das Datenbankpaßwort ein und drücken Sie SQL_Login. Wenn das Paßwort korrekt war oder nicht benötigt wird erscheint unten in der Ausgabe: "Login erfolgreich". Die restlichen Befehle sind jetzt freigegeben. Die Abfrage ist mit "SELECT * FROM mitglieder" und maximal 1 Datensatz vorbelegt. Sie ist für die Datenbank "VEREIN.MDB" im Verzeichnis "Datenbank" im Entwicklerpaket vorgesehen. Passen Sie die Abfrage für andere Datenbanken entsprechend an. Ist der Befehl gesendet, so bekommen Sie eine Request-ID zurück, mit der die Abfragen durchnummeriert werden. Sie können zu dieser ID weitere Datensätze abfragen, indem sie SQL-Next drücken. SQL_End terminiert die Abfrage, sofern noch nicht alle Daten ausgelesen sind. Nach einigen Minuten werden "herrenlose" Abfragen ebenfalls terminiert.



Siehe auch

[Programmierbeispiele](#) | [SQL](#)

[Send feedback to TAPIMaster@](#)
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3 Server-Schnittstelle

Die Server-Schnittstelle könnte in eine Datenbank wie Access, in einer Excel-Tabelle oder eine Anwendung eingebettet werden, welche die Funktion einer Datenbank hat. Das könnte auch ein Dienst sein, der letztendlich den Datenbankzugriff erledigt. Normalerweise jedoch können Sie die Anbindung an eine Datenbank über die ODBC-Schnittstelle des CTI-Servers erreichen.

Wie auf der Clientseite gibt es auch hier wieder zwei Schnittstellen in einer DLL. Diese enthält allerdings deutlich weniger Funktionen.

[Befehlsreferenz](#)

[Programmierbeispiele](#)

Siehe auch

[Inhalt](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1 Befehlsreferenz

Die Befehlsreferenz ist in die beiden Schnittstellen für ActiveX und C++API unterteilt. Die C++ Funktionen haben ein CTI_ in den Funktionen vorangestellt. Ansonsten sind Funktionen und Events in beiden Schnittstellen weitgehend identisch.

[.NET](#)
[ActiveX](#)
[C++ CTI API](#)

Siehe auch

[Serverschnittstelle](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1 .NET

Die Server-Schnittstelle besteht aus diesen vier Klassen. Die Klasse Telephony wird zwingend gebraucht, da hier die Initialisierung der Schnittstelle stattfindet.

ClientCom	Die Klasse dient dem Senden von Nachrichten von Server zum Client.und umgekehrt.
CtiConvert	Die Klasse dient der Umwandlung von Rufnummern und Datumsformaten.
PBXInfo	Die Klasse liefert Informationen zur Telefonanlage und den angemeldeten Clients.
Telephony	Die Klasse enthält alle Funktionen und Events welche mit den Gesprächen direkt zu tun haben.

Siehe auch

[Befehlsreferenz Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.1 ClientCom

Klasse ClientCom

Die Klasse dient dem Senden von Nachrichten von Server zum Client.und umgekehrt.

Funktionen

OnUserToDatabase	Benutzerdefinierte Daten werden zur weiteren Bearbeitung empfangen.
SendAdminMessage	Der Server sendet Infos an den Client.
SendIdentity	Eine Rufnummernidentifikation wird zurückgesendet.
SendUserData	Es werden benutzerdefinierte Daten gesendet.

Siehe auch

[Übersicht .NET](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.1.1 OnUserToDatabase

Benutzerdefinierte Daten werden zur weiteren Bearbeitung empfangen.

```
Event OnUserToDataBase  
(  
    object sender,  
    uint e.dwCommand,  
    String e.strUser,  
    String e.strMessage  
)
```

Parameter

sender

Objekt [ClientCom](#)

e.dwCommand

Nummer des Befehls

e.strUser

Durchwahl des Benutzers, der den Befehl abgeschickt hat.

e.strMessage

Parameter zu dem Befehl

Bemerkungen

Ein Client hat einen benutzerdefinierten Befehl an die Serverschnittstelle geschickt. Dies kann auf dem Client mit [UserSendToDataBase](#) erfolgen. Der Server handelt den Event ab und kann mit [SendUserData](#) an den Client antworten. Die Parameter *lCommand* und *strUser* werden für die Antwort an den Client benötigt.

Siehe auch

[Übersicht ClientCom](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.1.2 SendAdminMessage

Der Server sendet Infos an den Client.

```
void SendAdminMessage  
(  
    String strUser,  
    String strMessage  
)
```

Parameter

strUser

Durchwahl des Benutzers

strText

Text, welcher der Server schickt.

Rückgabewert

keiner

Bemerkungen

Damit können Textnachrichten vom Server zum Client geschickt werden, welche dann beim Client ans Nachricht angezeigt werden.

Siehe auch

[Übersicht ClientCom](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.1.3 SendIdentity

Eine Rufnummernidentifikation wird zurückgesendet.

```
void SendIdentity  
(  
    uint dwCallID,  
    String strUser,  
    String strText  
)
```

Parameter

dwCallID

Anruf-ID des Gesprächs

strUser

Durchwahl des Benutzers

strText

Text, welcher die Daten zum Anrufer enthält.

Rückgabewert

keiner

Bemerkungen

Die Serverschnittstelle hat ein Event [OnIdentity](#) erhalten. Daraufhin wurden Daten zu der Rufnummer ermittelt. Die Anruf-ID und Durchwahl des Teilnehmers sind mit OnGetIdentity übertragen worden und werden jetzt mit zurückgesendet. Der *strText* sollte so formatiert sein dass er vom jeweiligen Client angezeigt werden kann.

Siehe auch

[Übersicht ClientCom](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.1.4 SendUserData

Es werden benutzerdefinierte Daten gesendet.

```
void SendUserData  
(  
    uint dwCommand,  
    String strUser,  
    String strText  
)
```

Parameter

dwCommand
Nummer des Befehls

strUser
Durchwahl des Benutzers, der den Befehl abgeschickt hat.

strText
Parameter zu dem Befehl

Rückgabewert

keiner

Bemerkungen

TAPIMaster® ermöglicht das Senden von benutzerdefinierten Daten zwischen Client- und Serverschnittstelle und zurück. Wenn Sie zusätzliche Daten zwischen Client und Server übertragen wollen können Sie das hiermit tun ohne zwischen Client und Server eine zusätzliche Netzwerkverbindung aufmachen zu müssen. Mit [UserSendToDataBase](#) sendet der Client Befehle an den Server, welche dort den Event [OnUserToDataBase](#) auslösen.

Siehe auch

[Übersicht ClientCom](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.2 CtiConvert

Klasse CtiConvert

Die Klasse dient der Umwandlung von Rufnummern und Datumsformaten.

Funktionen

ConvertToCanonical	Umwandlung von Rufnummern in das kanonische Format.
GetDateTime	Wandelt time_t in Standardstring um.
GetFormatDateTime	Wandelt time_t in individuell formatierten String um.

Siehe auch

[Übersicht .NET](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.2.1 ConvertToCanonical

Umwandlung von Rufnummern in das kanonische Format.

```
String ConvertToCanonical  
(  
    String strInput  
)
```

Parameter

strInput

Rufnummer, die in das kanonische Format umgewandelt werden soll.

Rückgabewert

Umgewandelte Rufnummer

Bemerkungen

Sie können diese Funktion benutzen um Datenbestände mit Rufnummern in das kanonische Format umzuwandeln. Der Ausgabestring sollte mindestens 32 Zeichen groß sein.

Siehe auch

[Übersicht CtiConvert](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.2.2 GetDateTime

Wandelt time_t in Standardstring um.

```
String GetDateTime  
{  
    uint dwTime  
}
```

Parameter

ITime

Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.

Rückgabewert

String in einem Standardformat: 2007-01-12T00:53:32.875

Bemerkungen

Nutzen Sie diese Funktion wenn Sie ein gut sortierbares Format brauchen. Viele Programme wie Excel nutzen dieses Format.

Siehe auch

[Übersicht CtiConvert](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.2.3 GetFormatDateTime

Wandelt time_t in individuell formatierten String um.

```
String GetFormatDateTime
(
    uint dwTime,
    String strFormat
)
```

Parameter

dwTime

Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.

strFormat

String mit Datumsformat. Dieses kann aus den folgenden Formatzeichen gebildet werden.

Zeichen	Erklärung
%a	Abgekürzter Wochentag
%A	Ausgeschriebener Wochentag
%b	Abgekürzter Monatsname
%B	Ausgeschriebener Monatsname
%c	Lokales Datum und Uhrzeit, abhängig von Systemeinstellungen
%d	Monatstag (01 – 31)
%H	Stunde im 24Stunden-Format (00 – 23)
%I	Stunde im 12Stunden-Format (01 – 12)
%j	Tag im Jahr (001 – 366)
%m	Monat als Zahl (01 – 12)
%M	Minute (00 – 59)
%p	Lokaler Indikator für vormittag / nachmittag bei 12stündiger Anzeige
%S	Sekunden (00 – 59)
%U	Kalenderwoche, beginnend mit Sonntag (00 – 53)
%w	Wochentag als Zahl, beginnend mit Sonntag (0 – 6)
%W	Kalenderwoche, beginnend mit Montag (00 – 53)
%x	Lokales Datum, abhängig vom System
%X	Lokale Zeit, abhängig vom System
%y	2stellige Jahresangabe (00 – 99)
%Y	4stellige Jahresangabe
%Z, %z	Zeitzone
%%	Prozentzeichen

Rückgabewert

String in einem individuellen Format.

Bemerkungen

Die Formatzeichen können beliebig kombiniert werden.

Siehe auch

[Übersicht CtiConvert](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3 PBXInfo

Klasse PBXInfo

Die Klasse liefert Informationen zur Telefonanlage und den angemeldeten Clients.

Funktionen

GetExternPrefix	Vorwahl für Externgespräche.
GetInboundFilter	Ziffern, welche bei eingehenden Gesprächen gefiltert werden sollen.
GetInternPrefix	Vorwahl für Interngespräche.
GetNameFromExtension	Benutzername einer Nebenstelle.
GetOutboundFilter	Ziffern, welche bei ausgehenden Gesprächen gefiltert werden sollen.
LicenseCount	Liefert die Anzahl der Lizenzen zurück.
OnLoginData	Ein Benutzer hat sich ein- oder ausgeloggt.
TMServiceStatus	Zustand des TAPIMaster Dienstes.

Siehe auch

[Übersicht .NET](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.1 GetExternPrefix

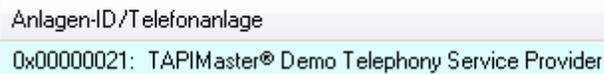
Vorwahl für Externgespräche.

```
String GetExternPrefix  
(  
    String strPBX  
)
```

Parameter

strPBX

Bezeichnung der Telefonanlage. Diese kann den CTI Server Einstellungen entnommen werden.



Anlagen-ID/Telefonanlage
0x00000021: TAPIMaster® Demo Telephony Service Provider

Rückgabewert

Diese Ziffer wird bei externen Gesprächen vorgewählt.

Bemerkungen

Das Ergebnis kann leer sein wenn keine Ziffer vorgewählt werden muss.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.2 GetInboundFilter

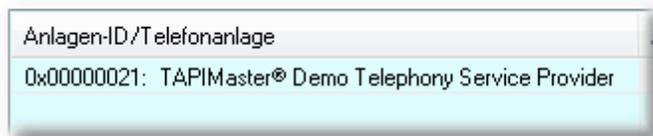
Ziffern, welche bei eingehenden Gesprächen gefiltert werden sollen.

```
String GetInboundFilter  
(  
    String strPBX  
)
```

Parameter

strPBX

Bezeichnung der Telefonanlage. Diese kann den TAPIMaster Server Einstellungen entnommen werden.

**Rückgabewert**

Die gefilterten Ziffern bei eingehenden Anrufen.

Bemerkungen

Das Ergebnis kann leer sein wenn nichts gefiltert wird.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.3 GetInternPrefix

Vorwahl für Interngespräche.

```
String GetInternPrefix  
(  
    String strPBX  
)
```

Parameter

strPBX

Bezeichnung der Telefonanlage. Diese kann den TAPIMaster Server Einstellungen entnommen werden.

**Rückgabewert**

Diese Ziffer wird bei internen Gesprächen vorgewählt.

Bemerkungen

Das Ergebnis kann leer sein wenn keine Ziffer vorgewählt werden muss.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.4 GetNameFromExtension

Benutzername einer Nebenstelle.

```
String GetNameFromExtension  
(  
    String strExtension  
)
```

Parameter

strExtension

Nebenstelle eines angemeldeten Benutzers.

Rückgabewert

Der Benutzername welcher zu der Nebenstelle gehört.

Bemerkungen

Der Benutzer muss sich unter dieser Nebenstelle mindestens einmal angemeldet haben.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.5 GetOutboundFilter

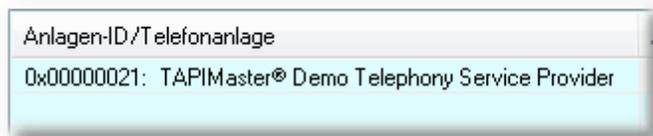
Ziffern, welche bei ausgehenden Gesprächen gefiltert werden sollen.

```
String GetOutboundFilter  
(  
    String strPBX  
)
```

Parameter

strPBX

Bezeichnung der Telefonanlage. Diese kann den TAPIMaster Server Einstellungen entnommen werden.

**Rückgabewert**

Die gefilterten Ziffern bei ausgehenden Anrufen.

Bemerkungen

Das Ergebnis kann leer sein wenn nichts gefiltert wird.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.6 LicenseCount

Liefert die Anzahl der Lizenzen zurück.

Eigenschaft LicenseCount

Typ

int, Nur lesen

Bemerkungen

Die Anzahl ist 0 wenn es sich nicht um eine Lizenzversion handelt oder die Lizenz zeitlich begrenzt war und abgelaufen ist.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.7 OnLoginData

Ein Benutzer hat sich ein- oder ausgeloggt.

```
Event OnLoginData
(  
    object sender,  
    String e.strDevice  
    String e.strName,  
    bool e.bLogin  
)
```

Parameter

sender

Objekt [PBXInfo](#)

e.strDevice

Nebenstelle des Benutzers, der sich ein- oder ausgeloggt hat.

e.strName

Anmeldename sofern bekannt.

e.bLogin

1 User hat sich eingeloggt.

0 User hat sich ausgeloggt.

Bemerkungen

Wird nach dem Einloggen eines Clients gesendet.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.3.8 TMServiceStatus

Zustand des TAPIMaster Dienstes.

Eigenschaft TMServiceStatus**Typ**

uint, Nur lesen. Bedeutung:

- 0 Unknown
- 1 SERVICE_STOPPED
- 2 SERVICE_START_PENDING
- 3 SERVICE_STOP_PENDING
- 4 SERVICE_RUNNING
- 5 SERVICE_CONTINUE_PENDING
- 6 SERVICE_PAUSE_PENDING
- 7 SERVICE_PAUSED

Bemerkungen

Die zurückgelieferten Konstanten entsprechen denen in winsvc.h.

Siehe auch

[Übersicht PBXInfo](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4 Telephony

Klasse Telephony

Die Klasse enthält alle Funktionen und Events welche mit den Gesprächen direkt zu tun haben.

Funktionen

Alternate	Wechselt zwischen zwei Gesprächspartnern hin und her.
Answer	Nimmt einen eingehenden Anruf an.
Conference	Verknüpft zwei Gespräche zu einer Konferenz.
Forward	Schaltet eine Rufumleitung.
GetPBXMonitoredLines	Listet alle überwachten Leitungen auf.
Hangup	Beendet ein Gespräch.
Init	Initialisiert die Serverschnittstelle.
MakeCall	Tätigt eine Anwahl vom Server aus.
OnCallNumber	Eine Rufnummer wird übertragen.
OnCallState	Übermittelt den Zustand eigener Gespräche.
OnForwardingState	Die Rufumleitung hat sich geändert.
OnIdentity	Zu einer Rufnummer wird eine Rufnummernidentifikation angefordert.
OnNewCall	Eine Vorabinfo zum Anruf kommt.
OnProtocolData	Ein Gesprächsdatensatz soll abgespeichert werden.
Pickup	Holt ein Gespräch von einem anderen Telefon heran.
Redirect	Leitet ein Gespräch zu einem anderen Telefon weiter.
SendAllUserCallStates	Liefert die aktuellen Callstates zurück.
SendLastCallRecords	Liefert die Daten der letzten Anrufe als Event zurück.
SendPBXLogins	Liefert die angemeldeten User zurück.
SetPhoneDisplay	Zeigt Text im Telefondisplay an.
Transfer	Verbindet zwei Gespräche. Der Akteur verabschiedet sich.

Siehe auch

[Übersicht .NET](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.1 Alternate

Wechselt zwischen zwei Gesprächspartnern hin und her.

```
void Alternate  
(  
    String strDevice  
)
```

Parameter

strDevice

Nebenstelle des Teilnehmers der zwischen den Anrufen wechseln soll.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.2 Answer

Nimmt einen eingehenden Anruf an.

```
void Answer  
(  
    String strDevice  
)
```

Parameter

strDevice

Nebenstelle des Benutzers der einen Anruf annehmen soll.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.3 Conference

Verknüpft zwei Gespräche zu einer Konferenz.

```
void Conference  
(  
    String strDevice  
)
```

Parameter

strDevice

Nebenstelle des Benutzers der eine Konferenz einleiten soll.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.4 Forward

Schaltet eine Rufumleitung.

```
void Forward  
(  
    String strDevice,  
    String strTraget  
)
```

Parameter

strDevice

Nebenstelle des Benutzers der eine Rufumleitung schaltet.

strTarget

Ziel der Rufumleitung.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.5 GetPBXMonitoredLines

Listet alle überwachten Leitungen auf.

```
String GetPBXMonitoredLines  
(  
    int nMaxCount  
)
```

Parameter

nMaxCount
Maximale Anzahl der Leitungen.

Rückgabewert

Liste mit den Durchwahlen, durch Komma getrennt.

Bemerkungen

Das sind alle Durchwahlen welche von der Lizenz unterstützt werden.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.6 Hangup

Beendet ein Gespräch.

```
void Hangup  
(  
    String strDevice  
)
```

Parameter

strDevice

Nebenstelle des Benutzers der einen Anruf auflegen soll.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.7 Init

Initialisiert die Serverschnittstelle.

void Init()

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Diese Funktion initialisiert die .NET-Schnittstelle in der FCTIDATA.DLL. Sie wird aufgerufen bevor die Schnittstelle benutzt wird.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.8 MakeCall

Tätigt eine Anwahl vom Server aus.

```
void MakeCall  
(  
    String strDevice,  
    String strTarget  
)
```

Parameter

strDevice

Nebenstelle des Benutzers der anrufen soll.

strTarget

Anzuwählende Nummer

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.9 OnCallNumber

Eine Rufnummer wird übertragen.

```
Event OnCallNumber
(  
    object sender,  
    String e.strDevice  
    uint e.dwCallID  
    String e.strNumber  
)
```

Parameter

sender

Objekt [Telephony](#)

e.strDevice

Nebenstelle des Benutzers, zu dem die Rufnummer des Gesprächspartners übertragen wird.

e.dwCallID

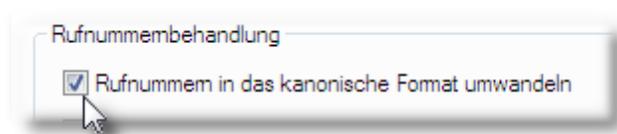
Anruf-ID

e.strNumber

Übertragene Rufnummer, maximal 31 Stellen.

Bemerkungen

Übermittelt Rufnummern, sofern diese mit übertragen wurden. Bei Anrufen aus dem Analognetz und anonymen Anrufen wird keine Rufnummer übertragen. Bei ausgehenden Rufen wird dieser Event ebenfalls ausgelöst. Das Format der Rufnummer kann auf dem Server als kanonisch festgelegt werden (Seite Anlageneinstellungen).



Normalerweise macht ein neuer Anruf auf sich aufmerksam, indem ein Event [OnCallState](#) mit dem Status [LINECALLSTATE_OFFERING](#) (eingehend) oder [LINECALLSTATE_DIALTONE](#) (ausgehend) auftritt. **OnCallNumber** kann jedoch auch vor allen anderen Events gesendet werden. Die Rufnummer kann sich während des Gesprächs ändern, wenn z. B. jemand dem Benutzer einen Anruf weitervermittelt.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.10 OnCallState

Übermittelt den Zustand eigener Gespräche.

```
Event OnCallState  
(  
    object sender,  
    String e.strDevice,  
    uint e.dwState,  
    uint e.dwCallID  
)
```

Parameter

sender

Objekt [Telephony](#)

e.strDevice

Nebenstelle des Benutzers, dessen Gesprächszustände übertragen werden.

e.dwState

[Gesprächszustand](#), in dem sich der Anruf befindet.

e.dwCallID

Anruf-ID, dessen Zustand sich ändert.

Bemerkungen

Der Zustand eines Gesprächs hat sich geändert.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.11 OnForwardingState

Die Rufumleitung hat sich geändert.

```
Event OnForwardingState
(  
    object sender,  
    String e.strDevice,  
    uint e.bSet,  
    uint e.strTarget  
)
```

Parameter

sender

Objekt [Telephony](#)

e.strDevice

Nebenstelle des Benutzers, von dem die Rufumleitung übertragen wird.

e.bSet

true wenn Rufumleitung aktiv ist und false wenn nicht.

e.strTarget

Ziel der Rufumleitung, kann auch dann Text enthalten wenn Umleitung inaktiv ist.

Bemerkungen

Die Rufumleitung hat sich geändert.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.12 OnIdentity

Zu einer Rufnummer wird eine Rufnummernidentifikation angefordert.

```
Event OnIdentity  
(  
    object sender,  
    uint e.dwCallID,  
    String e.strDevice,  
    String e.strNumber  
)
```

Parameter

sender

Objekt [Telephony](#)

e.dwCallID

Anruf-ID des Gesprächs

e.strDevice

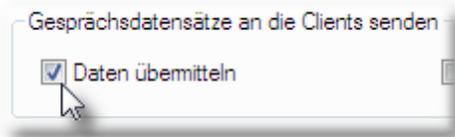
Durchwahl welche angerufen wurde oder anruft.

e.strNumber

Rufnummer, welche der andere Telefonpartner hat.

Bemerkungen

Kommt ein Gespräch zustande und wird eine Rufnummer erkannt, so wird dieser Event ausgelöst. Bedingung dafür ist eine zugeschaltete Rufnummernübermittlung an die Clients. In der Serverkonsole kann diese auf der Seite "Datenbankschnittstelle" geschaltet werden.



Auf diese Nachricht hin ermitteln Sie auf der Serverseite die Angaben zu der Rufnummer. Die Parameter *ICallID* und *strUser* werden für die Antwort benötigt. Die Rücksendung der Daten an den Client erfolgt dann mit [SendIdentity](#).

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.13 OnNew Call

Eine Vorabinfo zum Anruf kommt.

```
Event OnNewCall
(  
    object sender,  
    String e.strDevice,  
    uint e.dwCallID,  
    boole. blnbound,  
    uint e.dwStart,  
    String e.strNumber  
)
```

Parameter

sender

Objekt [Telephony](#)

e.strDevice

Durchwahl des Benutzers

e.dwCallID

Anruf-ID, dessen Zustand sich ändert.

e.blbound

Ist True wenn der Anruf eingehend war.

e.dwStart

Gesprächsbeginn als time_t

e.strNumber

Rufnummer des Gesprächspartners

Bemerkungen

Die Informationen werden zu Gesprächsbeginn gesendet.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.14 OnProtocolData

Ein Gesprächsdatensatz soll abgespeichert werden.

```
Event OnProtocolData
(  
    object sender,  
    uint e.dwStart,  
    uint e.dwEnd,  
    String e.strDevice,  
    String e.strNumber,  
    bool e.bSuccess,  
    bool e.bInbound,  
    String e.strAddress  
)
```

Parameter

sender

Objekt [Telephony](#)

e.dwStart

Gesprächsbeginn als time_t

e.dwEnd

Gesprächsende als time_t

e.strDevice

Nebenstelle des Benutzers

e.strNumber

Rufnummer des Gesprächspartners

e.bSuccess

Ist True wenn der Anruf erfolgreich war.

e.bInbound

Ist True wenn der Anruf eingehend war.

e.strAddress

Adresse, welche angerufen wurde. Ist bei eingehenden Anrufen von Bedeutung, wenn ein Telefon auch mehrere interne Durchwahlen (Adressen) anspricht. Das Telefon könnte für verschiedene Projekte oder Profitcenter nach außen hin mit diversen Rufnummern verknüpft sein, anhand der Adresse lässt sich dann zuordnen für welches Profitcenter welche Telefonleistung erbracht wurde.

Bemerkungen

Wurde ein Anruf mit einer Rufnummer erkannt, so wird dieser Event ausgelöst. Die Serverschnittstelle kann die Daten daraufhin speichern.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.15 Pickup

Holt ein Gespräch von einem anderen Telefon heran.

```
void Pickup  
(  
    String strDevice,  
    String strRingin  
)
```

Parameter

strDevice

Nebenstelle des Benutzers der ein Gespräch heran holt.

strRingin

Nebenstelle des Telefons von dem ein Gespräch heran geholt werden soll.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.16 Redirect

Leitet ein Gespräch zu einem anderen Telefon weiter.

```
void Redirect  
(  
    String strDevice,  
    String strTarget  
)
```

Parameter

strDevice

Nebenstelle des Benutzers der ein Gespräch weiterleitet.

strTarget

Durchwahl des Telefons zu dem das Gespräch weitergeleitet werden soll.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.17 SendAllUserCallStates

Liefert die aktuellen Callstates zurück.

```
void SendAllUserCallStates()
```

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Auf diese Funktion hin werden [OnCallState](#) Events generiert.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.18 SendLastCallRecords

Liefert die Daten der letzten Anrufe als Event zurück.

```
void SendLastCallRecords(  
    String strDevice,  
    int nMaxCount,  
    bool bDelete  
)
```

Parameter

strDevice

Durchwahl des Benutzers von dem die Anrufinformationen benötigt werden.

nMaxCount

Maximale Anzahl der Datensätze welche gesendet werden sollen.

bDelete

Sagt ob der Anrufpuffer für den Benutzer gelöscht werden soll.

Rückgabewert

keiner

Bemerkungen

Die Daten werden über Events vom Typ [OnProtocolData](#) gesendet.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.19 SendPBXLoginLines

Liefert die angemeldeten User zurück.

```
void SendPBXLoginLines()
```

Parameter

Keine

Rückgabewert

keiner

Bemerkungen

Auf diese Funktion hin werden [OnLoginData](#) Events generiert.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.20 SetPhoneDisplay

Zeigt Text im Telefondisplay an.

```
void SetPhoneDisplay  
(  
    String strDevice,  
    String strText  
)
```

Parameter

strDevice

Nebenstelle wo Text angezeigt werden soll.

strText

Angezeigter Text.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen. Nur wenige Anlagen unterstützen diese Funktion.

Siehe auch

[Übersicht Telephony](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.1.4.21 Transfer

Verbindet zwei Gespräche. Der Akteur verabschiedet sich.

```
void Transfer  
(  
    String strDevice  
)
```

Parameter

strDevice

Durchwahl des Teilnehmers der zwei Anrufe verbinden soll.

Rückgabewert

keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

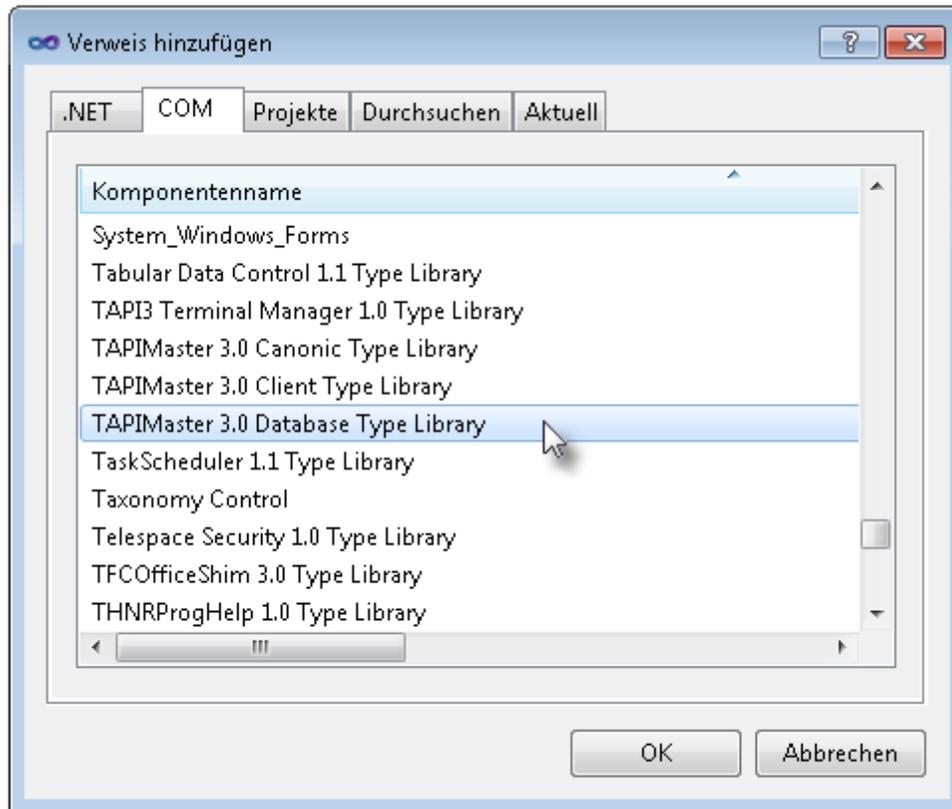
[Übersicht Telephony](#)

[Send feedback to TAPIMaster®](#)

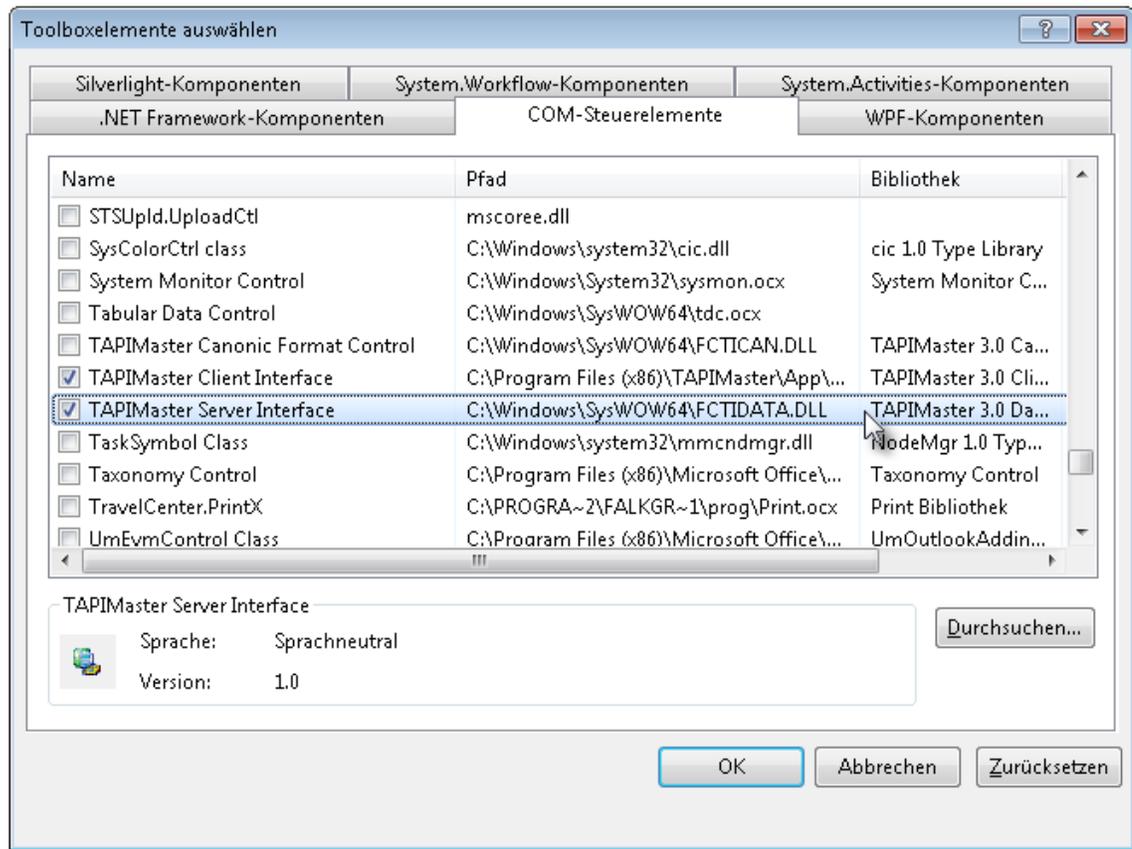
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2 ActiveX

Die Schnittstelle für ActiveX befindet sich in der Datei FCTIDATA.DLL. Diese muss registriert sein. Binden Sie die entsprechenden Verweise in Ihr Projekt mit ein. Führen Sie auf dem Zielrechner ein Serversetup aus, damit dort alle notwendigen Dateien vorhanden bzw. registriert sind. Hier die Vorgehensweise für die Einbindung in VB / C#: Projekt | Verweis hinzufügen...



Jetzt kann das Objekt in die Toolbox übernommen werden. Anschließend wird es auf dem Formular platziert.



Die Verwendung der Schnittstelle können Sie im Beispiel für die [Einbindung in Excel und VB](#) sehen.

[Funktionen](#)

[Events](#)

Siehe auch

[Befehlsreferenz Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1 Funktionen

Initialisierung

InitCOM	Initialisiert die Serverschnittstelle.
-------------------------	--

Datenaustausch

SendIdentity	Eine Rufnummernidentifikation wird zurückgesendet.
SendUserData	Es werden benutzerdefinierte Daten gesendet.

Telefoniefunktionen

Alternate	Wechselt zwischen zwei Gesprächspartnern hin und her.
Answer	Anruf annehmen.
Conference	Verknüpft zwei Gespräche zu einer Konferenz.
Drop	Anruf auflegen.
MakeCall	Anruf einleiten
Pickup	Holt ein Gespräch von einem anderen Telefon heran.
Redirect	Leitet ein Gespräch zu einem anderen Telefon weiter.
Transfer	Verbindet zwei Gespräche. Der Akteur verabschiedet sich.

Zusatzfunktionen

ConvertToCanonical	Wandelt eine Rufnummer in das kanonische Format um.
GetAllUserCallStates	Liefert die aktuellen Gespräche aller überwachten Leitungen als Events zurück.
GetDateTime	Wandelt time_t in Standardstring um.
GetExtension	Interne Funktion
GetExternPrefix	Vorwahl für Externgespräche.
GetFilterCalibrationState	Interne Funktion
GetFormatDateTime	Wandelt time_t in individuell formatierten String um.
GetInboundFilter	Ziffern, welche bei eingehenden Gesprächen gefiltert werden sollen.
GetInternPrefix	Vorwahl für Interngespräche
GetLastCallRecords	Liefert die Daten der letzten Anrufe als Event zurück.
GetOutboundFilter	Ziffern, welche bei ausgehenden Gesprächen gefiltert werden sollen.

GetPBXFiltersAndPrefixes	Liefert Filter und Vorwahlen zurück.
GetPBXMonitoredLines	Listet alle überwachten Leitungen auf.
LicenseCount	Liefert die Anzahl der CTI-Lizenzen zurück.
ResetCalibration	Interne Funktion.
SetCalibrationParams	Interne Funktion.
SetPhoneDisplay	Interne Funktion.
TMServiceStatus	Zustand des TAPIMaster Dienstes
UserCount	Interne Funktion.
UserNameFromNumber	Liefert den letzten angemeldeten Benutzer zu einer Durchwahl zurück.

Nicht aufgeführte Funktionen sind veraltet oder nur für den internen Gebrauch bestimmt. Diese sollten nicht verwendet werden.

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.1 Alternate

Wechselt zwischen zwei Gesprächspartnern hin und her.

```
Funktion Alternate(  
    String bstrUser  
)
```

Parameter

bstrUser

Durchwahl des Teilnehmers der zwischen den Anrufen wechseln soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.2 Answer

Nimmt einen eingehenden Anruf an.

```
Funktion Answer(  
    String bstrUser  
)
```

Parameter

bstrUser

Durchwahl des Benutzers der einen Anruf annehmen soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.3 Conference

Verknüpft zwei Gespräche zu einer Konferenz.

```
Funktion Conference(  
    String bstrUser  
)
```

Parameter

bstrUser

Durchwahl des Benutzers der eine Konferenz einleiten soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.4 ConvertToCanonical

Umwandlung von Rufnummern in das kanonische Format.

```
Funktion ConvertToCanonical(  
    String bstrUser  
)
```

```
String bstrInput,  
StringPtr pbstrOutput  
)
```

Parameter*bstrInput*

Rufnummer, die in das kanonische Format umgewandelt werden soll.

pbstrOutput

Umgewandelte Rufnummer

Bemerkungen

Sie können diese Funktion benutzen um Datenbestände mit Rufnummern in das kanonische Format umzuwandeln. Der Ausgabestring sollte mindestens 32 Zeichen groß sein.

Beispiel[ServerInterface](#)**Siehe auch**[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.5 Drop

Beendet ein Gespräch.

```
Funktion Drop(  
String bstrUser  
)
```

Parameter*bstrUser*

Durchwahl des Benutzers der einen Anruf auflegen soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel[ServerInterface](#)**Siehe auch**[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.6 GetAllUserCallStates

Liefert die aktuellen Gespräche aller überwachten Leitungen als Events zurück.

Function GetAllUserCallStates()**Parameter***keine***Bemerkungen**

Zu jedem Anruf wird ein Event [OnCallState](#) und [OnNewCall](#) geliefert.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.7 GetDateTime

Wandelt time_t in Standardstring um.

```
Function GetDateTime(
    Long ITime,
    StringPtr pbstrTimeString,
)
```

Parameter*ITime*

Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.

pbstrTimeString

Nullterminierter String in einem Standardformat: 2007-01-12T00:53:32.875

Bemerkungen

Nutzen Sie diese Funktion wenn Sie ein gut sortierbares Format brauchen. Viele Programme wie Excel nutzen dieses Format.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.8 GetExtension

Interne Funktion.

[Send feedback to TAPIMaster®](#)

<%FULLCOPYRIGHT%

1.3.1.2.1.9 GetExternPrefix

Vorwahl für Externgespräche.

```
Function GetExternPrefix(
```

```
String bstrPBX,
StringPtr pbstrPrefix,
)
```

Parameter*bstrPBX*

Bezeichnung der Telefonanlage. Diese kann den CTI Server Einstellungen entnommen werden.

*pbstrPrefix*

Diese Ziffer wird bei externen Gesprächen vorgewählt.

Bemerkungen

Das Ergebnis kann leer sein wenn keine Ziffer vorgewählt werden muss.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.10 GetFilterCalibrationState

Interne Funktion.

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.11 GetFormatDateTime

Wandelt time_t in individuell formatierten String um.

```
Function GetFormatDateTime(
    Long ITime,
    String bstrFormat,
    StringPtr pbstrTimeString,
)
```

Parameter*ITime*

Wert im Format time_t, welcher z. B. im Event [OnProtocolData](#) zurückgeliefert wird.

bstrFormat

String mit Datumsformat. Dieses kann aus den folgenden Formatzeichen gebildet werden.

Zeichen	Erklärung
---------	-----------

%a	Abgekürzter Wochentag
%A	Ausgeschriebener Wochentag
%b	Abgekürzter Monatsname
%B	Ausgeschriebener Monatsname
%c	Lokales Datum und Uhrzeit, abhängig von Systemeinstellungen
%d	Monatstag (01 – 31)
%H	Stunde im 24Stunden-Format (00 – 23)
%I	Stunde im 12Stunden-Format (01 – 12)
%j	Tag im Jahr (001 – 366)
%m	Monat als Zahl (01 – 12)
%M	Minute (00 – 59)
%p	Lokaler Indikator für vormittag / nachmittag bei 12stündiger Anzeige
%S	Sekunden (00 – 59)
%U	Kalenderwoche, beginnend mit Sonntag (00 – 53)
%w	Wochentag als Zahl, beginnend mit Sonntag (0 – 6)
%W	Kalenderwoche, beginnend mit Montag (00 – 53)
%x	Lokales Datum, abhängig vom System
%X	Lokale Zeit, abhängig vom System
%y	2stellige Jahresangabe (00 – 99)
%Y	4stellige Jahresangabe
%z,%Z	Zeitzone
%%	Prozentzeichen

pbstrTimeString

Nullterminierter String in einem individuellen Format.

Bemerkungen

Die Formatzeichen können beliebig kombiniert werden.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.12 GetInboundFilter

Ziffern, welche bei eingehenden Gesprächen gefiltert werden sollen.

```
Function GetInboundFilter(
    String bstrPBX,
    StringPtr pbstrFilter,
)
```

Parameter*bstrPBX*

Bezeichnung der Telefonanlage. Diese kann den TAPIMaster Server Einstellungen entnommen werden.

*pbstrFilter*

Die gefilterten Ziffern bei eingehenden Anrufen.

Bemerkungen

Das Ergebnis kann leer sein wenn nichts gefiltert wird.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.13 GetInternPrefix

Vorwahl für Interngespräche.

```
Function GetInternPrefix(  
    String bstrPBX,  
    StringPtr pbstrPrefix,  
)
```

Parameter*bstrPBX*

Bezeichnung der Telefonanlage. Diese kann den TAPIMaster Server Einstellungen entnommen werden.

*pbstrPrefix*

Diese Ziffer wird bei internen Gesprächen vorgewählt.

Bemerkungen

Das Ergebnis kann leer sein wenn keine Ziffer vorgewählt werden muss.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.14 GetLastCallRecords

Liefert die Daten der letzten Anrufe als Event zurück.

```
Function GetLastCallRecords(  
    String bstrUSER,  
    Long IMaxCount,  
    Bool bDelete  
)
```

Parameter

bstrUser

Durchwahl des Benutzers von dem die Anrufinformationen benötigt werden.

IMaxCount

Maximale Anzahl der Datensätze welche gesendet werden sollen.

bDelete

Sagt ob der Anrufpuffer für den Benutzer gelöscht werden soll.

Bemerkungen

Die Daten werden über Events vom Typ [OnProtocolData](#) gesendet.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.15 GetOutboundFilter

Ziffern, welche bei ausgehenden Gesprächen gefiltert werden sollen.

```
Function GetOutboundFilter(  
    String bstrPBX,  
    StringPtr pbstrFilter,  
)
```

Parameter

bstrPBX

Bezeichnung der Telefonanlage. Diese kann den TAPIMaster Server Einstellungen entnommen werden.



pbstrFilter

Die gefilterten Ziffern bei ausgehenden Anrufen.

Bemerkungen

Das Ergebnis kann leer sein wenn nichts gefiltert wird.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.16 GetPBXFiltersAndPrefixes

Liefert Filter und Vorwahlen zurück.

```
Function GetPBXFiltersAndPrefixes(  
    String bstrPBX,  
    StringPtr pbstrExtern,  
    StringPtr pbstrIntern,  
    StringPtr pbstrInbound,  
    StringPtr pbstrOutbound  
)
```

Parameter

bstrPBX

Bezeichnung der Telefonanlage. Diese kann den TAPIMaster Server Einstellungen entnommen werden.



pbstrExtern

Externvorwahl.

pbstrIntern

Internvorwahl.

pbstrInbound

Die gefilterten Ziffern bei eingehenden Anrufen.

pbstrOutbound

Die gefilterten Ziffern bei ausgehenden Anrufen.

Bemerkungen

Die Ergebnisse können ganz oder teilweise leer sein.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.17 GetPBXMonitoredLines

Listet alle überwachten Leitungen auf.

```
Function GetPBXMonitoredLines(  
    StringPtr pbstrLines  
)
```

Parameter

pbstrLines

Liste mit den Durchwahlen, durch Komma getrennt.

Bemerkungen

Das sind alle Durchwahlen welche von der Lizenz unterstützt werden.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.18 InitCOM

Initialisiert die Serverschnittstelle.

```
Funktion InitCOM()
```

Parameter

Keine

Bemerkungen

Diese Funktion initialisiert die COM-Schnittstelle in der FCTIDATA.DLL. Sie wird aufgerufen bevor die Schnittstelle benutzt wird.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.19 LicenseCount

Liefert die Anzahl der Lizenzen zurück.

```
Eigenschaft LicenseCount
```

Typ

Long, Nur lesen

Bemerkungen

Die Anzahl ist 0 wenn es sich nicht um eine Lizenzversion handelt oder die Lizenz zeitlich begrenzt war und abgelaufen ist.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.20 MakeCall

Tätigt eine Anwahl vom Server aus.

```
Funktion MakeCall(  
    String bstrUser,  
    String bstrNumber  
)
```

Parameter

bstrUser

Durchwahl des Benutzers der anrufen soll.

bstrNumber

Anzuvählende Nummer

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.21 Pickup

Holt ein Gespräch von einem anderen Telefon heran.

```
Funktion Pickup(  
    String bstrUser,  
    String bstrNumber  
)
```

Parameter

bstrUser

Durchwahl des Benutzers der ein Gespräch heran holt.

bstrNumber

Durchwahl des Telefons von dem ein Gespräch heran geholt werden soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.22 Redirect

Leitet ein Gespräch zu einem anderen Telefon weiter.

```
Funktion Redirect(  
    String bstrUser,  
    String bstrNumber  
)
```

Parameter

bstrUser

Durchwahl des Benutzers der ein Gespräch weiterleitet.

bstrNumber

Durchwahl des Telefons zu dem das Gespräch weitergeleitet werden soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.23 ResetCalibration

Interne Funktion.

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.24 SendIdentity

Eine Rufnummernidentifikation wird zurückgesendet.

```
Funktion SendIdentity(  
    Long ICallID,  
    String bstrUser,  
    String bstrText  
)
```

Parameter

ICallID

Anruf-ID des Gesprächs

bstrUser

Durchwahl des Benutzers

bstrText

Text, welcher die Daten zum Anrufer enthält.

Bemerkungen

Die Serverschnittstelle hat ein Event [OnGetIdentity](#) erhalten. Daraufhin wurden Daten zu der Rufnummer ermittelt. Die Anruf-ID und Durchwahl des Teilnehmers sind mit [OnGetIdentity](#) übertragen worden und werden jetzt mit zurückgesendet. Der *bstrText* sollte so formatiert sein dass er vom jeweiligen Client angezeigt werden kann.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#) | [OnIdentity](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.25 SendUserData

Es werden benutzerdefinierte Daten gesendet.

```
Funktion SendUserData(  
    Long ICommand,  
    String bstrUser,  
    String bstrText  
)
```

Parameter

ICommand

Nummer des Befehls

bstrUser

Durchwahl des Benutzers, der den Befehl abgeschickt hat.

bstrText

Parameter zu dem Befehl

Bemerkungen

TAPIMaster® ermöglicht das Senden von benutzerdefinierten Daten zwischen Client- und Serverschnittstelle und zurück. Wenn Sie zusätzliche Daten zwischen Client und Server übertragen wollen können Sie das hiermit tun ohne zwischen Client und Server eine zusätzliche Netzwerkverbindung aufmachen zu müssen. Mit [EUserSendToDataBase](#) sendet der Client Befehle an den Server, welche dort den Event [OnUserToDataBase](#) auslösen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.26 SetCalibrationParams

Interne Funktion.

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.27 SetPhoneDisplay

Interne Funktion.

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.28 TMServiceStatus

Zustand des TAPIMaster Dienstes.

Eigenschaft TMServiceStatus

Typ

Long, Nur lesen. Bedeutung:

- 0 Unknown
- 1 SERVICE_STOPPED
- 2 SERVICE_START_PENDING
- 3 SERVICE_STOP_PENDING
- 4 SERVICE_RUNNING
- 5 SERVICE_CONTINUE_PENDING
- 6 SERVICE_PAUSE_PENDING
- 7 SERVICE_PAUSED

Bemerkungen

Die zurückgelieferten Konstanten entsprechen denen in winsvc.h.

Siehe **auch**

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.29 Transfer

Verbindet zwei Gespräche. Der Akteur verabschiedet sich.

```
Funktion Transfer(  
    String bstrUser  
)
```

Parameter

bstrUser

Durchwahl des Teilnehmers der zwei Anrufe verbinden soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe **auch**

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.30 UserCount

Interne Funktion.

[Send feedback to TAPIMaster®](#)
 © 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.1.31 UserNameFromNumber

Liefert den letzten angemeldeten Benutzer zu einer Durchwahl zurück.

```
Function UserNameFromNumber(
    String bstrNumber,
    StringPtr pbstrUser,
)
```

Parameter

bstrNumber
 Durchwahl zu welcher der Benutzername gesucht wird.

pbstrUser
 Gefundener Benutzername.

Bemerkungen

Das Ergebnis kann leer sein wenn sich noch kein Benutzer angemeldet hat.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)
 © 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2 Events

Datenaustausch

OnCallNumber	Rufnummer wird übermittelt.
OnCallState	Anrufstatus wird übermittelt.
OnForwardingState	Die Rufumleitung hat sich geändert.
OnGetIdentity	Rufnummernidentifikation wird angefordert.
OnLoginData	Ein Benutzer hat sich ein- oder ausgeloggt.
OnNewCall	Vorabinformationen zu einem Anruf werden übermittelt.
OnProtocolData	Gesprächsdatensätze werden zum Abspeichern übergeben.
OnUserToDataBase	Benutzerdefinierte Daten werden gesendet.

Nicht aufgeführte Events sind veraltet oder nur für den internen Gebrauch bestimmt. Diese sollten nicht verwendet werden.

Siehe auch

[Übersicht ActiveX](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.1 OnCallNumber

Eine Rufnummer wird übertragen.

```
Event OnCallNumber(  
    String bstrUser  
    Long ICallID  
    String bstrNumber  
)
```

Parameter

bstrUser

Durchwahl des Benutzers, zu dem die Rufnummer des Gesprächspartners übertragen wird.

ICallID

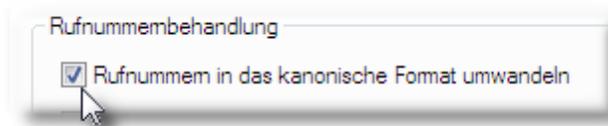
Anruf-ID

bstrNumber

Übertragene Rufnummer, maximal 31 Stellen.

Bemerkungen

Übermittelt Rufnummern, sofern diese mit übertragen wurden. Bei Anrufen aus dem Analognetz und anonymen Anrufen wird keine Rufnummer übertragen. Bei ausgehenden Rufen wird dieser Event ebenfalls ausgelöst. Das Format der Rufnummer kann auf dem Server als kanonisch festgelegt werden (Seite Anlageneinstellungen).



Normalerweise macht ein neuer Anruf auf sich aufmerksam, indem ein Event [OnCallState](#) mit dem Status [LINECALLSTATE_OFFERING](#) (eingehend) oder [LINECALLSTATE_DIALTONE](#) (ausgehend) auftritt. **OnCallNumber** kann jedoch auch vor allen anderen Events gesendet werden. Die Rufnummer kann sich während des Gesprächs ändern, wenn z. B. jemand dem Benutzer einen Anruf weitervermittelt.

Siehe auch

[Übersicht Events Server](#) | [OnCallState](#) | [Gesprächszustände](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.2 OnCallState

Übermittelt den Zustand eigener Gespräche.

```
Event OnCallState(  
    String bstrUser  
    Long IState,  
    Long ICallID  
)
```

Parameter

bstrUser

Durchwahl des Benutzers, dessen Gesprächszustände übertragen werden.

IState

[Gesprächszustand](#), in dem sich der Anruf befindet.

ICallID

Anruf-ID, dessen Zustand sich ändert.

Bemerkungen

Der Zustand eines Gesprächs hat sich geändert.

Siehe auch

[Übersicht Events Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.3 OnForwardingState

Die Rufumleitung hat sich geändert.

```
Event OnForwardingState(  
    String bstrUser  
    Bool bSet,  
    String bstrTarget  
)
```

Parameter

bstrUser

Nebenstelle des Benutzers, von dem die Rufumleitung übertragen wird.

bSet

true wenn Rufumleitung aktiv ist und false wenn nicht.

bstrTarget

Ziel der Rufumleitung, kann auch dann Text enthalten wenn Umleitung inaktiv ist.

Bemerkungen

Die Rufumleitung wurde geändert.

Siehe auch[Übersicht Events Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.4 OnGetIdentity

Zu einer Rufnummer wird eine Rufnummernidentifikation angefordert.

```
Event OnGetIdentity(  
    Long ICallID,  
    String bstrUser,  
    String bstrNumber  
)
```

Parameter*ICallID*

Anruf-ID des Gesprächs

bstrUser

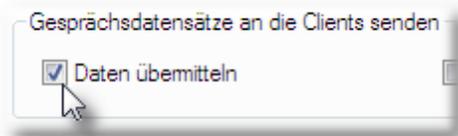
Durchwahl welche angerufen wurde oder anruft.

bstrNumber

Rufnummer, welche der andere Telefonpartner hat.

Bemerkungen

Kommt ein Gespräch zustande und wird eine Rufnummer erkannt, so wird dieser Event ausgelöst. Bedingung dafür ist eine zugeschaltete Rufnummernübermittlung an die Clients. In der Serverkonsole kann diese auf der Seite "Datenbankschnittstelle" geschaltet werden.



Auf diese Nachricht hin ermitteln Sie auf der Serverseite die Angaben zu der Rufnummer. Die Parameter *ICallID* und *bstrUser* werden für die Antwort benötigt. Die Rücksendung der Daten an den Client erfolgt dann mit [SendIdentity](#).

Siehe auch[Übersicht Events Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.5 OnLoginData

Ein Benutzer hat sich ein- oder ausgeloggt.

```
Event OnLoginData(  
    String bstrUser  
    String bstrName,  
    Bool bLogin  
)
```

Parameter

bstrUser

Durchwahl des Benutzers, der sich ein- oder ausgeloggt hat.

bstrName

Anmeldename sofern bekannt.

bLogin

1 User hat sich eingeloggt.

0 User hat sich ausgeloggt.

Bemerkungen**Siehe auch**

[Übersicht Events Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.6 OnNew Call

Eine Vorabinfo zum Anruf kommt.

```
Event OnNewCall(  
    String bstrUser,  
    Long ICallID,  
    Bool bInbound,  
    Long IStart,  
    String bstrNumber  
)
```

Parameter

bstrUser

Durchwahl des Benutzers

ICallID

Anruf-ID, dessen Zustand sich ändert.

bInbound

Ist True wenn der Anruf eingehend war.

IStart

Gesprächsbeginn als time_t

bstrNumber

Rufnummer des Gesprächspartners

Bemerkungen

Die Informationen werden zu Gesprächsbeginn gesendet.

Siehe auch

[Übersicht Events Server](#) | [GetDateTime](#) | [GetFormatDateTime](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.7 OnProtocolData

Ein Gesprächsdatensatz soll abgespeichert werden.

```
Event OnProtocolData(  
    Long IStart,  
    Long IEnd,  
    String bstrUser,  
    String bstrNumber,  
    Bool bSuccess,  
    Bool blnbound,  
    String bstrAddress  
)
```

Parameter

IStart

Gesprächsbeginn als time_t

IEnd

Gesprächsende als time_t

bstrUser

Durchwahl des Benutzers

bstrNumber

Rufnummer des Gesprächspartners

bSuccess

Ist True wenn der Anruf erfolgreich war.

blnbound

Ist True wenn der Anruf eingehend war.

bstrAddress

Adresse, welche angerufen wurde. Ist bei eingehenden Anrufen von Bedeutung, wenn ein Telefon auch mehrere interne Durchwahln (Adressen) anspricht. Das Telefon könnte für verschiedene Projekte oder Profitcenter nach außen hin mit diversen Rufnummern verknüpft sein, anhand der Adresse lässt sich dann zuordnen für welches Profitcenter welche Telefonleistung erbracht wurde.

Bemerkungen

Wurde ein Anruf mit einer Rufnummer erkannt, so wird dieser Event ausgelöst. Die Serverschnittstelle kann die Daten daraufhin speichern.

Siehe auch

[Übersicht Events Server](#) | [GetDateTime](#) | [GetFormatDateTime](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.2.2.8 OnUserToDataBase

Benutzerdefinierte Daten werden zur weiteren Bearbeitung empfangen.

```
Event OnUserToDataBase(  
    Long ICommand,  
    String bstrUser,  
    String bstrMessage  
)
```

Parameter

ICommand

Nummer des Befehls

bstrUser

Durchwahl des Benutzers, der den Befehl abgeschickt hat.

bstrMessage

Parameter zu dem Befehl

Bemerkungen

Ein Client hat einen benutzerdefinierten Befehl an die Serverschnittstelle geschickt. Dies kann auf dem Client mit [EUserSendToDataBase](#) erfolgen. Der Server handelt den Event ab und kann mit [SendUserData](#) an den Client antworten. Die Parameter *ICommand* und *bstrUser* werden für die Antwort an den Client benötigt.

Siehe auch

[Übersicht Events Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3 C++ CTI API

Die Schnittstelle für die C++ API befindet sich in der Datei FCTIDATA.DLL. Sie benötigen weiterhin die Dateien CTIDATA.H und CTITYPES.H sowie FCTIDATA.LIB aus dem CTI SDK. Kopieren Sie diese Dateien in die zugehörigen Verzeichnisse Ihrer Entwicklungsumgebung. Führen Sie auf dem Zielrechner ein Serversetup aus, damit dort alle notwendigen Dateien vorhanden bzw. registriert sind.

[Funktionen](#)

[Events](#)

Siehe auch

[Befehlsreferenz Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1 Funktionen

Präfix

Die Funktionen der Serverschnittstelle haben den Präfix "CTI_T" vorangestellt.

Initialisierung

CTI_Init	Initialisiert die Serverschnittstelle.
--------------------------	--

Datenaustausch

CTI_SendIdentity	Eine Rufnummernidentifikation wird zurückgesendet.
CTI_SendUserData	Es werden benutzerdefinierte Daten gesendet.

Zusatzfunktionen

CTI_ConvertToCanonical	Wandelt eine Rufnummer in das kanonische Format um.
CTI_GetLicenseCount	Liefert die Anzahl der CTI-Lizenzen zurück.
CTI_GetDateTime	Wandelt time_t in Standardstring um.
CTI_GetFormattedDateTime	Wandelt time_t in individuell formatierten String um.
CTI_MakeCall	Anruf einleiten.
CTI_Answer	Anruf annehmen.
CTI_Drop	Anruf auflegen.
CTI_Conference	Verknüpft zwei Gespräche zu einer Konferenz.
CTI_Pickup	Holt ein Gespräch von einem anderen Telefon heran.
CTI_Redirect	Leitet ein Gespräch zu einem anderen Telefon weiter.
CTI_Alternate	Wechselt zwischen zwei Gesprächspartnern hin und her.
CTI_Transfer	Verbindet zwei Gespräche. Der Akteur verabschiedet sich.

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster®](#)
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.1 CTI_Alternate

Wechselt zwischen zwei Gesprächspartnern hin und her.

```
VOID WINAPI CTI_Alternate(  
    PTCHAR pszUser  
)
```

Parameter

pszUser

Durchwahl des Teilnehmers der zwischen den Anrufen wechseln soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.2 CTI_Answer

Nimmt ein Gespräch entgegen.

```
VOID WINAPI CTI_Answer(  
    PTCHAR pszUser  
);
```

Parameter

pszUser

Durchwahl des Benutzers der ein Gespräch annehmen soll.

Rückgabewert

Keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.3 CTI_Conference

Verknüpft zwei Gespräche zu einer Konferenz.

```
VOID WINAPI CTI_Conference(  
    PTCHAR pszUser
```

)

Parameter

pszUser

Durchwahl des Benutzers der eine Konferenz einleiten soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.4 CTI_ConvertToCanonical

Umwandlung von Rufnummern in das kanonische Format.

```
VOID CTI_ConvertToCanonical(  
    PTCHAR pszInput,  
    PTCHAR pszOutput  
);
```

Parameter

pszInput

Rufnummer, die in das kanonische Format umgewandelt werden soll.

pszOutput

Umgewandelte Rufnummer

Rückgabewert

Keiner

Bemerkungen

Sie können diese Funktion benutzen um Datenbestände mit Rufnummern in das kanonische Format umzuwandeln. Der Ausgabestring sollte mindestens 32 Zeichen groß sein.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.5 CTI_Drop

Legt ein Gespräch auf.

```
VOID WINAPI CTI_Drop(  

```

PTCHAR pszUser

);

Parameter*pszUser*

Durchwahl des Benutzers der ein Gespräch auflegen soll.

Rückgabewert

Keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.6 CTI_Forward

Geben Sie hier den Text ein.

1.3.1.3.1.7 CTI_GetDateTime

Wandelt time_t in Standardstring um.

```
PTCHAR CTI_GetDateTime(  
    __time32_t wTime,  
);
```

Parameter*wTime*Wert im Format time_t, welcher z. B. im Event [CTI_PROTOCOLDATA](#) zurückgeliefert wird.**Rückgabewert**

Nullterminierter String in einem Standardformat: 2007-01-12T00:53:32.875

Bemerkungen

Nutzen Sie diese Funktion wenn Sie ein gut sortierbares Format brauchen. Viele Programme wie Excel nutzen dieses Format.

Siehe auch[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.8 CTI_GetFormatDateTime

Wandelt time_t in individuell formatierten String um.

```
PTCHAR CTI_GetFormatDateTime(
    __time32_t wTime,
    PTCHAR pszFormat
);
```

Parameter

wTime

Wert im Format time_t, welcher z. B. im Event [CTI_PROTOCOLDATA](#) zurückgeliefert wird.

pszFormat

String mit Datumsformat. Dieses kann aus den folgenden Formatzeichen gebildet werden.

Zeichen	Erklärung
%a	Abgekürzter Wochentag
%A	Ausgeschriebener Wochentag
%b	Abgekürzter Monatsname
%B	Ausgeschriebener Monatsname
%c	Lokales Datum und Uhrzeit, abhängig von Systemeinstellungen
%d	Monatstag (01 – 31)
%H	Stunde im 24Stunden-Format (00 – 23)
%I	Stunde im 12Stunden-Format (01 – 12)
%j	Tag im Jahr (001 – 366)
%m	Monat als Zahl (01 – 12)
%M	Minute (00 – 59)
%p	Lokaler Indikator für vormittag / nachmittag bei 12stündiger Anzeige
%S	Sekunden (00 – 59)
%U	Kalenderwoche, beginnend mit Sonntag (00 – 53)
%w	Wochentag als Zahl, beginnend mit Sonntag (0 – 6)
%W	Kalenderwoche, beginnend mit Montag (00 – 53)
%x	Lokales Datum, abhängig vom System
%X	Lokale Zeit, abhängig vom System
%y	2stellige Jahresangabe (00 – 99)
%Y	4stellige Jahresangabe
%z,%Z	Zeitzone
%%	Prozentzeichen

Rückgabewert

Nullterminierter String in einem individuellen Format.

Bemerkungen

Die Formatzeichen können beliebig kombiniert werden.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.9 CTI_GetLicenseCount

Liefert die Anzahl der Lizenzen zurück.

```
LONG WINAPI CTI_GetLicenseCount();
```

Parameter

keine

Rückgabewert

Anzahl der Lizenzen

Bemerkungen

Die Anzahl ist 0 wenn es sich nicht um eine Lizenzversion handelt oder die Lizenz zeitlich begrenzt war und abgelaufen ist.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.10 CTI_GetNameFromExtension

Geben Sie hier den Text ein.

1.3.1.3.1.11 CTI_GetPBXLoginLines

Geben Sie hier den Text ein.

1.3.1.3.1.12 CTI_GetPBXMonitoredLines

Geben Sie hier den Text ein.

1.3.1.3.1.13 CTI_Init

Initialisiert die Serverschnittstelle.

```
VOID WINAPI CTI_Init(  
    DataCallbackProc pCallbackProc  
);
```

Parameter

pCallbackProc

Adresse einer [Callback](#)-Funktion

Rückgabewert

Keiner

Bemerkungen

Diese Funktion initialisiert die Server-Schnittstelle in der FCTIDATA.DLL. Sie wird aufgerufen bevor die Schnittstelle benutzt wird.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.14 CTI_MakeCall

Tätigt eine Anwahl vom Server aus.

```
VOID WINAPI CTI_MakeCall(  
    PTCHAR pszUser,  
    PTCHAR pszNumber  
);
```

Parameter

pszUser

Durchwahl des Benutzers der anrufen soll.

pszNumber

Anzuwählende Nummer

Rückgabewert

Keiner

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.15 CTI_Pickup

Holt ein Gespräch von einem anderen Telefon heran.

```
VOID WINAPI CTI_Pickup(  
    PTCHAR pszUser,  
    PTCHAR pszNumber  
)
```

Parameter

pszUser

Durchwahl des Benutzers der ein Gespräch heran holt.

pszNumber

Durchwahl des Telefons von dem ein Gespräch heran geholt werden soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.16 CTL_Redirect

Leitet ein Gespräch zu einem anderen Telefon weiter.

```
VOID WINAPI Redirect(  
    PTCHAR pszUser,  
    PTCHAR pszNumber  
)
```

Parameter

pszUser

Durchwahl des Benutzers der ein Gespräch weiterleitet.

pszNumber

Durchwahl des Telefons zu dem das Gespräch weitergeleitet werden soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster@](#)
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.17 CTL_SendAdminMessage

Geben Sie hier den Text ein.

1.3.1.3.1.18 CTL_SendIdentity

Eine Rufnummernidentifikation wird zurückgesendet.

```
VOID WINAPI CTL_SendIdentity(  
    LONG ICallID,  
    PTCHAR pszUser,  
    PTCHAR pszText  
);
```

Parameter

ICallID

Anruf-ID des Gesprächs

pszUser

Durchwahl des Benutzers

pszText

Text, welcher die Daten zum Anrufer enthält.

Rückgabewert

Keiner

Bemerkungen

Die Serverschnittstelle hat ein Event [CTI_GETIDENTITY](#) erhalten. Daraufhin wurden Daten zu der Rufnummer ermittelt. Die Anruf-ID und Durchwahl des Teilnehmers sind mit [CTI_GETIDENTITY](#) übertragen worden und werden jetzt mit zurückgesendet. Der *pszText* sollte so formatiert sein dass er vom jeweiligen Client angezeigt werden kann.

Siehe auch

[Send feedback to TAPIMaster®](#)

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.19 CTI_SendUserData

Es werden benutzerdefinierte Daten gesendet.

```
VOID WINAPI CTI_SendUserData(  
    LONG ICommand,  
    PTCHAR pszUser,  
    PTCHAR pszText  
);
```

Parameter

ICommand

Nummer des Befehls

pszUser

Durchwahl des Benutzers, der den Befehl abgeschickt hat.

pszText

Parameter zu dem Befehl

Rückgabewert

Keiner

Bemerkungen

TAPIMaster® ermöglicht das Senden von benutzerdefinierten Daten zwischen Client- und Serverschnittstelle und zurück. Wenn Sie zusätzliche Daten zwischen Client und Server übertragen wollen können Sie das hiermit tun ohne zwischen Client und Server eine zusätzliche Netzwerkverbindung aufmachen zu müssen. Mit [CTI_EUserSendToDataBase](#) sendet der Client Befehle an den Server, welche dort den Event [CTI_DATABASETOUSER](#) auslösen.

Siehe auch

[Send feedback to TAPIMaster®](#)

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.1.20 CTI_SetPhoneDisplay

Geben Sie hier den Text ein.

1.3.1.3.1.21 CTI_Transfer

Verbindet zwei Gespräche. Der Akteur verabschiedet sich.

```
VOID WINAPI CTI_Transfer(  
    PTCHAR pszUser  
)
```

Parameter

pszUser

Durchwahl des Teilnehmers der zwei Anrufe verbinden soll.

Bemerkungen

Für den Benutzer muss eine Lizenz vorliegen.

Beispiel

[ServerInterface](#)

Siehe auch

[Übersicht Funktionen Server](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2 Events

Übermittlung

Alle Events kommen über eine Callback-Funktion herein. Sie ist folgendermaßen definiert:

```
typedef VOID (__stdcall* DataCallbackProc)(DWORD,LPVOID);
```

Im DWORD ist der unten aufgeführte Typ enthalten, LPVOID zeigt auf eine Struktur welche funktionsabhängig entweder ein [CTI_COMMAND_MSG](#) oder [CTI_PHONE_DATA](#) sein kann.

Typen

Parameter	Wert	Beschreibung
CTI_CALLNUMBER	0x00012001	Eine Rufnummer wird übertragen.
CTI_CALLSTATE	0x00030002	Übermittelt den Zustand von Gesprächen.
CTI_FORWARDINGSTATE	0x00012004	Die Rufumleitung hat sich geändert.
CTI_IDENTITY	0x0001200D	Zu einer Rufnummer wird eine Rufnummernidentifikation angefordert.
CTI_NEWCALL	0x00012018	Vorabinformationen zu einem Anruf kommen.
CTI_PROTOCOLDATA	0x0001200B	Ein Gesprächsdatensatz soll abgespeichert werden.
CTI_USERTODATABASE	0x00013007	Benutzerdefinierte Daten werden zur weiteren Bearbeitung empfangen.

Siehe auch

[Übersicht C++ CTI API](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2.1 CTI_CALLNUMBER

Eine Rufnummer wird übertragen.

Nachricht CTI_CALLNUMBER (0x00012001)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_IParam

Die Anruf-ID ist hier gespeichert.

m_unNumber

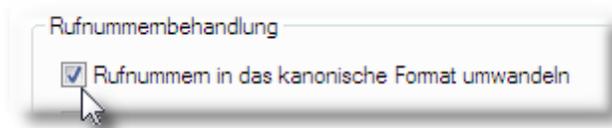
Durchwahl des Benutzers, zu dem die Rufnummer des Gesprächspartners übertragen wird.

m_szMessage

Rufnummer des anderen Teilnehmers.

Bemerkungen

Die Nachricht wird nur geschickt wenn tatsächlich Rufnummern übermittelt wurden. Bei einem ausgehenden Gespräch sollte das immer der Fall sein. Es kann passieren dass diese Nachricht die erste Information zu einem Anruf ist, d. h. es wurden noch keine Nachrichten vom Typ [CTI_CALLSTATE](#) vorher geschickt. Die Rufnummer kann sich während des Gesprächs ändern, wenn z. B. jemand dem Benutzer einen Anruf weitervermittelt. Das Format der Rufnummer kann auf dem Server als kanonisch festgelegt werden (Seite Anlageneinstellungen).



Siehe auch

[Übersicht Events \(Server\)](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2.2 CTI_CALLSTATE

Übermittelt den Zustand von Gesprächen.

Nachricht CTI_CALLSTATE (0x00030002)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

[Gesprächszustand](#), in dem sich der Anruf befindet.

m_lParam

Anruf-ID des Gesprächs

m_unNumber

Durchwahl des Benutzers, von dem die Gesprächszustände übertragen werden.

Bemerkungen

Der Zustand eines Gesprächs hat sich geändert.

Siehe auch

[Übersicht Events \(Server\)](#) | [Gesprächszustände](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2.3 CTI_FORWARDINGSTATE

Die Rufumleitung hat sich geändert.

Nachricht CTI_FORWARDINGSTATE (0x00010004)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_unNumber

Durchwahl des Benutzers, von dem die Rufumleitung übertragen wird.

m_wParam

1 wenn Rufumleitung aktiv ist und 0 wenn nicht.

m_szMessage

Ziel der Rufumleitung, kann auch dann Text enthalten wenn Umleitung inaktiv ist.

Bemerkungen

Die Rufumleitung wurde geändert.

Siehe auch

[Übersicht Events \(Server\)](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2.4 CTI_IDENTITY

Zu einer Rufnummer wird eine Rufnummernidentifikation angefordert.

Nachricht CTI_IDENTITY (0x0001200D)**Nachrichtentyp**

CTI_COMMAND_MSG

Benutzte Werte

m_IParam

Anruf-ID des Gesprächs

m_unNumber

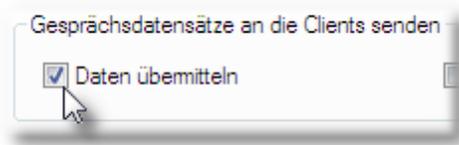
Durchwahl welche angerufen wurde oder anruft.

m_szMessage

Rufnummer, welche der andere Telefonpartner hat.

Bemerkungen

Kommt ein Gespräch zustande und wird eine Rufnummer erkannt, so wird dieser Event ausgelöst. Bedingung dafür ist eine zugeschaltete Rufnummernübermittlung an die Clients. In der Serverkonsole kann diese auf der Seite "Datenbankschnittstelle" geschaltet werden.



Auf diese Nachricht hin ermitteln Sie auf der Serverseite die Angaben zu der Rufnummer. Die Rücksendung der Daten an den Client erfolgt dann mit [CTI_SendIdentity](#).

Siehe auch

[Übersicht Events \(Server\)](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2.5 CTI_NEWCALL

Vorabinformationen zu einem Anruf kommen.

Nachricht CTI_NEWCALL (0x0001200B)

Nachrichtentyp

CTI_COMMAND_MSG

Benutzte Werte

m_wParam

Gesprächsbeginn als time_t

m_lParam

Anruf-ID des Gesprächs

m_unNumber

Durchwahl des Benutzers.

m_unOther

wNumber ist true wenn eingehendes Gespräch vorliegt.

m_szMessage

Rufnummer des anderen Teilnehmers.

Bemerkungen

Liefert eine Vorabinfo mit wichtigen Daten des Gesprächs.

Siehe auch

[Übersicht Events \(Server\)](#) | [CTI_GetDateTime](#) | [CTI_GetFormatDateTime](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2.6 CTI_PROTOCOLDATA

Ein Gesprächsdatensatz soll abgespeichert werden.

Nachricht CTI_PROTOCOLDATA (0x0001200B)**Nachrichtentyp**

CTI_PHONE_DATA

Benutzte Werte

m_dwStart

Gesprächsbeginn als time_t

m_dwEnd

Gesprächsende als time_t

m_unActor

Durchwahl des Benutzers

m_szNumber

Rufnummer seines Gesprächspartners

m_dwFlags

Bitflag, die Bits bedeuten:

0x00000002 Gespräch war erfolgreich

0x00000004 Gespräch war eingehend

m_szMessage

Adresse, welche angerufen wurde. Ist bei eingehenden Anrufen von Bedeutung, wenn ein Telefon auch mehrere interne Durchwahlen (Adressen) anspricht. Das Telefon könnte für verschiedene Projekte oder Profitcenter nach außen hin mit diversen Rufnummern verknüpft sein, anhand der Adresse lässt sich dann zuordnen für welches Profitcenter welche Telefonleistung erbracht wurde.

Bemerkungen

Wurde ein Anruf mit einer Rufnummer erkannt, so wird dieser Event ausgelöst. Die Serverschnittstelle kann die Daten daraufhin speichern.

Siehe auch

[Übersicht Events \(Server\)](#) | [CTI_GetDateTime](#) | [CTI_GetFormatDateTime](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.1.3.2.7 CTI_USERTODATABASE

Benutzerdefinierte Daten werden zur weiteren Bearbeitung empfangen.

Nachricht CTI_USERTODATABASE (0x00013007)**Nachrichtentyp**

CTI_PHONE_DATA

Benutzte Werte

m_IParam

Nummer des Befehls

m_unNumber

Durchwahl des Benutzers, der den Befehl abgeschickt hat.

m_szMessage

Parameter zu dem Befehl

Bemerkungen

Ein Client hat einen benutzerdefinierten Befehl an die Serverschnittstelle geschickt. Dies kann auf dem Client mit [CTI_EUserSendToDataBase](#) erfolgen. Der Server handelt den Event ab und kann mit [CTI_SendUserData](#) an den Client antworten.

Siehe auch

[Übersicht Events \(Server\)](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.2 Programmierbeispiele

Sie finden hier ein C++ Beispiel.

[Anwahl über IP-Schnittstelle](#)

[ServerCommands](#)

[ServerInterface](#)

Siehe auch

[Serverschnittstelle](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

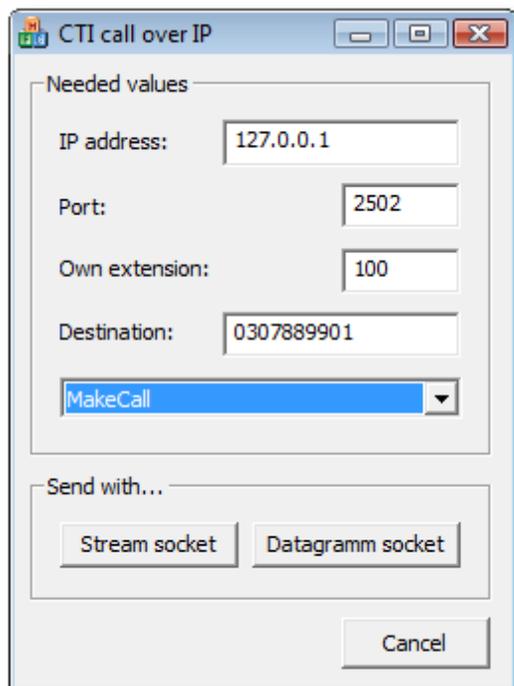
1.3.2.1 Anwahl über IP-Schnittstelle

In der Lizenzversion ist eine IP-Schnittstelle enthalten. Sie können Anwahlbefehle an den Server schicken und so auch aus anderen Betriebssystemen heraus wählen. Die Anwahlbefehle haben das Format:

MakeCall <Anrufer> <Nummer>

z. B. MakeCall 301 01191

(Der Apparat 301 soll die Zeitanzeige anrufen.)



Den Quelltext für die Anwahl auf Windows-BS können Sie dem Beispiel IPSEnd entnehmen.

Siehe auch

[Programmierbeispiele](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.2.2 ServerCommands

Benötigte Dateien

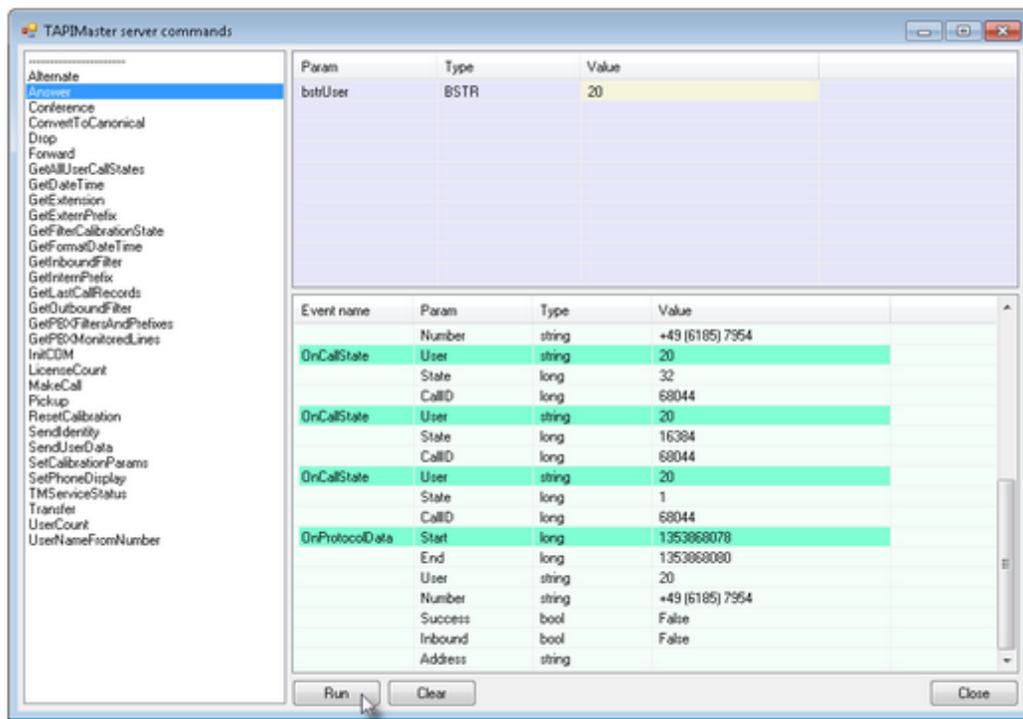
Installieren Sie den Server.

Quelltext

Der Quelltext des Programms ist im SDK enthalten

Beispiel

Mit Hilfe des Programms kann jeder Befehl der Server Schnittstelle einzeln aufgerufen werden. Das Programm ist ähnlich dem [CTI Browser](#), nur für die Server Schnittstelle.



Siehe auch

[Programmierbeispiele](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.3.2.3 ServerInterface

Benötigte Dateien

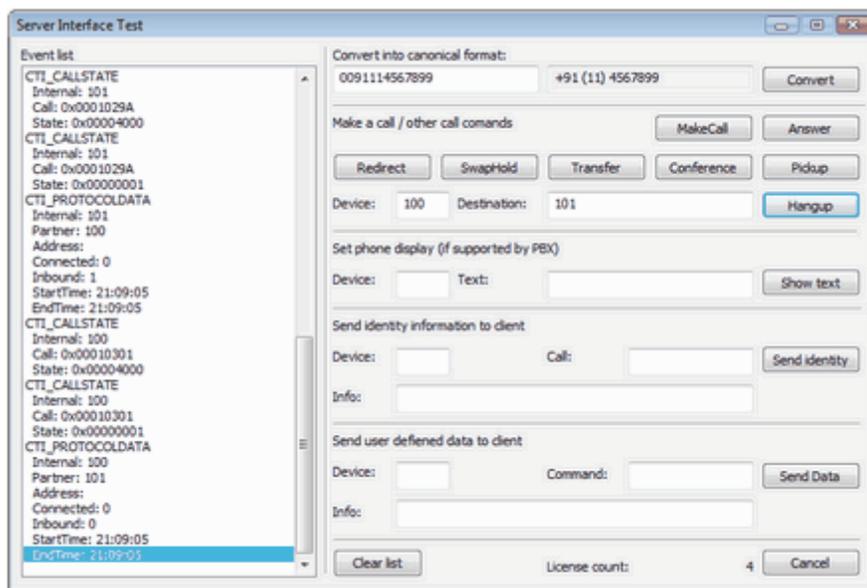
Installieren Sie den Server.

Quelltext

Der Quelltext des Programms ist im SDK enthalten

Beispiel

Das Beispielprogramm bindet die C++ Funktionen der Serverschnittstelle ein. Es zeigt Events des Servers an. Mit einer Hilfsfunktion kann eine Rufnummer in das kanonische Format formatiert werden. Die Serverschnittstelle beherrscht einfache Funktionen wie MakeCall, Answer und Drop. Das Programm liegt auch als VB und C# Version vor.



Siehe auch

[Programmierbeispiele](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.4 Setups

Allgemeines

Die Installation von CTI-Schnittstellen auf mehreren hundert Clients ist eine Sache die keinem Administrator zugemutet werden kann. Deshalb gibt es bei TAPIMaster® einige Möglichkeiten, diese Prozedur stark zu vereinfachen. Nicht in jedem Fall ist eine Installation der Clients notwendig. Sowohl Client- als auch Server-Setup lassen sich in andere Setups einbinden. Mit dem [Setup-Konfigurator](#) können Sie die Installationen vorkonfigurieren.

[Clients ohne Installation verwenden](#)

[Eingebettetes Client-Setup](#)

[Eingebettetes Server-Setup](#)

[Maßgeschneidertes Server-Setup](#)

[Installationskonfiguration](#)

[Deinstallation](#)

Siehe auch

[Inhalt](#)

[Send feedback to TAPIMaster®](#)

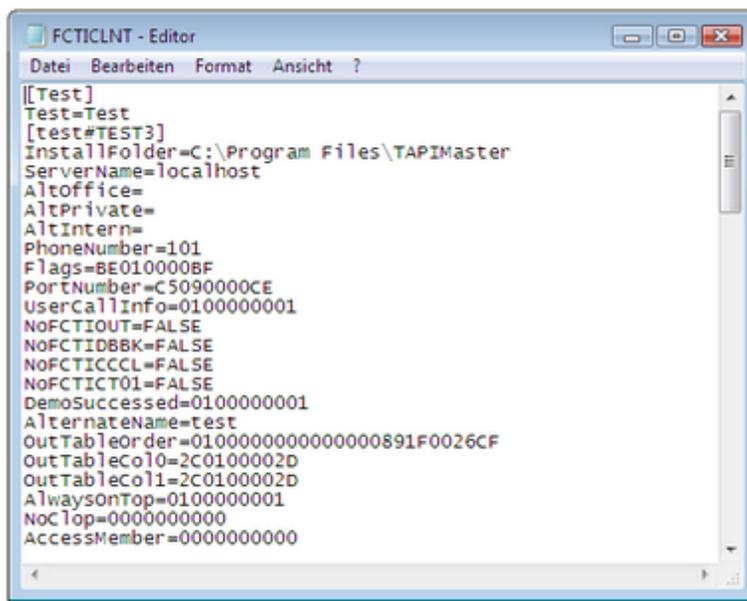
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.4.1 Clients ohne Installation verwenden

Die einfachste Möglichkeit des Setups ist natürlich dieses ganz zu vermeiden. Dazu gehen Sie so vor:

- Führen Sie das Setup auf dem CTI-Server durch.
- Im Ordner ..\Programme\TAPIMaster®\Setup finden Sie die zu Setups nötigen Dateien. Geben Sie dieses Verzeichnis im Netz frei oder kopieren Sie die Dateien in ein freigegebenes Verzeichnis.
- Wenn Sie einen eigenen Client geschrieben haben kopieren Sie ihn mit in das Verzeichnis. Dieser kann allerdings in dieser Variante keine [ActiveX-Schnittstelle](#) benutzen.
- Erstellen Sie einen Link auf den Standardclient bzw. Ihren Client, der im Netz eindeutig ist.
- Mailen Sie den Link an alle Benutzer.

Die Clients werden dann ohne Installation gestartet, beim ersten Start wird nach den [Verbindungsparametern](#) gefragt. Normalerweise speichern die Clients ihre Einstellungen lokal. Im vorliegenden Setup ist es aber sicher gewünscht, dass die Clients alle eine gemeinsame Konfigurationsdatei benutzen. Installieren Sie dazu einen Client lokal, nehmen Sie seine FCTICLNT.INI aus dem Windows-Verzeichnis und kopieren Sie die Datei in das freigegebene Netzverzeichnis. Die Clients benutzen dann nach dem Start diese Datei. Jeder Client benutzt dabei eine eigene Sektion, verschiedene Benutzer am selben Rechner benutzen verschiedene Sektionen, so dass sie ihre Einstellungen nicht gegenseitig überschreiben. Beispiel für eine FCTICLNT.INI:



```
[Test]
Test=Test
[test#TEST3]
InstallFolder=C:\Program Files\TAPIMaster
ServerName=localhost
AltOffice=
AltPrivate=
AltIntern=
PhoneNumber=101
Flags=BE010000BF
PortNumber=C5090000CE
UserCallInfo=0100000001
NoFCTIOUT=FALSE
NoFCTIDBBK=FALSE
NoFCTICCL=FALSE
NoFCTICT01=FALSE
DemoSucceeded=0100000001
AlternateName=test
OutTableOrder=0100000000000000891F0026CF
OutTableCol0=2C0100002D
OutTableCol1=2C0100002D
AlwaysOnTop=0100000001
NoClose=0000000000
AccessMember=0000000000
```

Siehe auch

[Setups](#) | [CTICLIENT_CHECKCONNECTION](#) | [CTI_OGetIniFileName](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.4.2 Eingebettetes Client-Setup

Normalerweise wird der Client über einen mehrseitigen Dialog installiert. Wenn die Clientinstallation jedoch Bestandteil einer größeren Installation sein soll ist das sehr unpraktisch. Sie können das Client-Setup in Ihr Setup so einbinden:

- Kopieren Sie die Dateien aus dem Verzeichnis `..\Programme\TAPIMaster®\Setup` in ein Unterverzeichnis in Ihrem Setup. Entfernen Sie nicht benötigte [Dateien](#). Fügen Sie ggf. Ihren Client hinzu.
- Im Verzeichnis befindet sich eine `FCTIINST.INI`. In dieser Datei sind die Einstellungen für das Setup enthalten. Bearbeiten Sie diese Datei mit dem [Setup-Konfigurator](#), so dass der Client mit den gewünschten Einstellungen gestartet wird. Aktivieren Sie "Automatisiertes Setup".
- Rufen Sie während der Installation die Datei `SETUP.EXE` auf, welche dann den Client installiert.

Siehe auch

[Setups](#) | [Setup-Konfigurator](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.4.3 Eingebettetes Server-Setup

Das Server-Setup lässt sich in ein größeres Setup einbetten, ohne dass der TAPIMaster® Setup-Dialog angezeigt wird. Gehen Sie dabei so vor:

- Kopieren Sie das Verzeichnis ..\Programme\TAPIMaster®\Setup in Ihre Installation. Entfernen Sie nicht benötigte [Dateien](#).
- Starten Sie zu Beginn Ihres Setups die Datei LW.EXE mit dem Parameter -I. Der Leitungsbeobachter wird nicht angezeigt und erstellt eine [Installationskonfiguration](#) für den Rechner.
- Zeigen Sie die Einstellungen für den Server in Ihrem Setup an. Vergleichen Sie dazu auch die Seite [Server-Setup](#) des [Setup-Konfigurators](#).
- Speichern Sie die vom Benutzer geänderten Werte wieder in die [Installationskonfiguration](#) ab.
- Starten Sie die SETUP.EXE mit dem Parameter -S. Der Server wird installiert und eingerichtet.

Siehe auch

[Setups](#) | [Setup-Konfigurator](#) | [Installationskonfiguration](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.4.4 Maßgeschneidertes Server-Setup

Nehmen wir folgenden Fall an: Ein Kunde hat Probleme mit der Server-Installation. Sie wollen ihn dabei unterstützen. Gehen Sie so vor:

- Der Kunde startet den [Leitungsbeobachter](#). Auf der Seite "Leistungsmerkmale der TAPI-Leitung" drückt er den Schalter "Liste speichern". Jetzt wird eine [Installationskonfiguration](#) erzeugt. Der Kunde mailt Ihnen diese Datei (FCTIINST.INI) zu. Daneben informiert er Sie über die internen Durchwahlen, die er verwenden möchte.
- Öffnen Sie die [Installationskonfiguration](#) mit dem [Setup-Konfigurator](#). Sie können die TAPI-Konfiguration des Kunden jetzt einsehen.
- Vergewissern Sie sich, dass auf dem Rechner des Kunden überhaupt ein Anlagentreiber installiert ist. Stellen Sie die gewünschte Anlage ein und aktivieren Sie die Rufnummern. Bei der Gelegenheit kann gleich die Lizenz vergeben werden.
- Bearbeiten Sie die Seite "Allgemeines" des [Setup-Konfigurators](#). "Automatisiertes Setup" sollte aktiviert sein.
- Erstellen Sie aus dem Verzeichnis eine selbstentpackende Datei, welche die SETUP.EXE aufruft.
- Der Kunde startet die Datei und der Server wird automatisch eingerichtet.

Siehe auch

[Setups](#) | [Setup-Konfigurator](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.4.5 Installationskonfiguration

In der Datei FCTIINST.INI sind die Installationseinstellungen enthalten. Die Datei wird bei der Installation des Programms ausgewertet und die voreingestellten Komponenten installiert. Die Datei kann vom [Leitungsbeobachter](#) und vom [Setup-Konfigurator](#) erzeugt werden. Beim Client-Setup haben Sie diese Einstellungen mit dem [Setup-Konfigurator](#) eingestellt. Beim [eingebetteten Server-Setup](#) sollten Sie die grün gefärbten Einstellungen während des Setups änderbar machen.

Einstellungen, welche für Client- und Server-Setup gelten.

```
[Default]
PortNumber=2501
AutoSetup=0
AutoSetupType=0
MaxIntern=2
CityCode=69
```

Einstellungen für das Client-Setup

```
[Client]
ServerName=PDC
Flags=414
InstallStandardClient=0
InstallOutlook=0
InstallDatabase=0
InstallUserDefined=0
InstallCOM=0
GenerateShortcuts=0
```

Für diese Einstellungen können Sie möglicherweise die Vorgaben verwenden wie hier voreingestellt.

```
[Server]
CanonicalFormat=1
SendCallData=1
SaveTextFile=1
SelfRepair=0
Pickup=1
Redirect=1
Remember=1
Forward=1
Chat=1
AddToGroup=1
License=
ComputerCode=494-113-823-124
```

Ist NumProvider Null, sollte die Server-Installation abgebrochen werden, da kein Anlagentreiber installiert ist.

```
[Provider]
Provider00=tiptel 195 Service Provider
Provider01=AGFEO TK-ServiceProvider3
NumProvider=2
```

```
SelectedProvider=0
```

Unten die Konfiguration der einzelnen Anlagen. Speichern Sie die Texte für die gewünschte Anlage wieder in dem Format {Leitungsname;Durchwahl;Aktivierungsflag} ab, da die Installation die Leitungen sonst nicht erkennt. Ändern Sie die Leitungsnamen nicht! Die Durchwahl kann bei manchen Anlagen fehlen.

```
[tiptel 195 Service Provider]
```

```
ProviderID=79
MaxNumActiveCalls=1
MaxNumConference=0
TransferModes=0
AddressFeatures=2
CallFeatures=2097348
CallFeatures2=0
NumLines=1
Line0000={tiptel phone1;;1}
```

```
[AGFEO TK-ServiceProvider3]
```

```
ProviderID=81
MaxNumActiveCalls=5
MaxNumConference=3
TransferModes=3
AddressFeatures=142
CallFeatures=235047149
CallFeatures2=0
NumLines=12
Line0000={Telefon 101;101;1}
Line0001={Telefon 102;102;1}
Line0002={Telefon 103;103;1}
Line0003={Telefon 104;104;0}
Line0004={Telefon 105;105;0}
Line0005={Telefon 106;106;0}
Line0006={Telefon 107;107;0}
Line0007={Telefon 20;20;0}
Line0008={Telefon 21;21;0}
Line0009={Telefon 31;31;1}
Line0010={Telefon 32;32;1}
Line0011={Telefon 33;33;1}
```

Siehe auch

[Setups](#) | [Setup-Konfigurator](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.4.6 Deinstallation

Sie können mit dem [Setup-Konfigurator](#) auch Programme erstellen welche Client und Server vom Rechner entfernen. Das geschieht dann ohne Rückfrage. Aktivieren Sie dazu auf der Seite "Allgemeines" des Konfigurators "Automatisiertes Setup" und schalten Sie "Deinstallation" ein. Die benötigten [Dateien](#)

entsprechen denen der Server-Installation. Generieren Sie mit Winzip eine selbstextrahierende Datei welche die Datei SETUP.EXE aufruft. Die Dateien werden entfernt.

Siehe auch

[Setups](#) | [Setup-Konfigurator](#)

[Send feedback to TAPIMaster@](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.5 Tools

Für die Entwicklung stehen verschiedene Hilfsmittel zur Verfügung. Mit dem CTI-Browser können Sie Befehle ausprobieren ohne programmieren zu müssen. Um ein CTI-Setup in Ihr Programmsetup mit einzubinden, nutzen Sie den Setup-Konfigurator.

[CTI-Browser](#)

[Leitungsbeobachter](#)

[Setup-Konfigurator](#)

Siehe auch

[Inhalt](#)

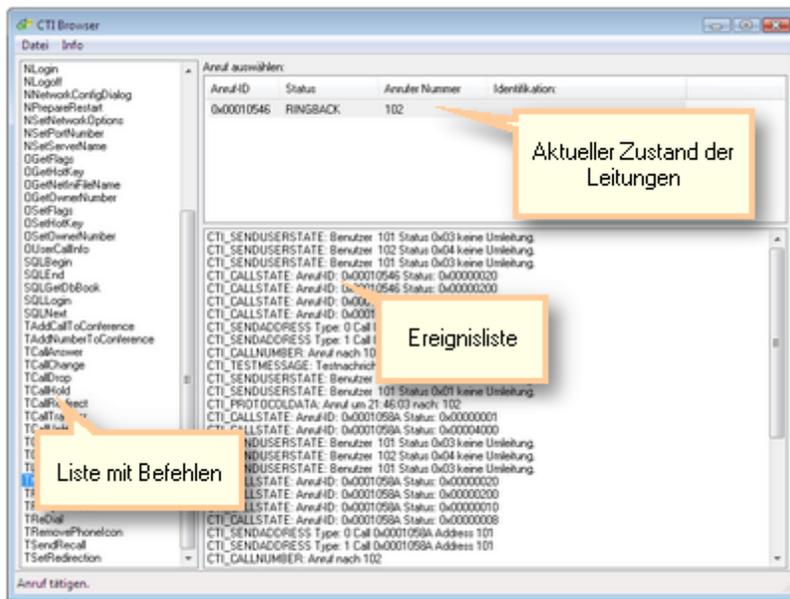
[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.5.1 CTI-Browser

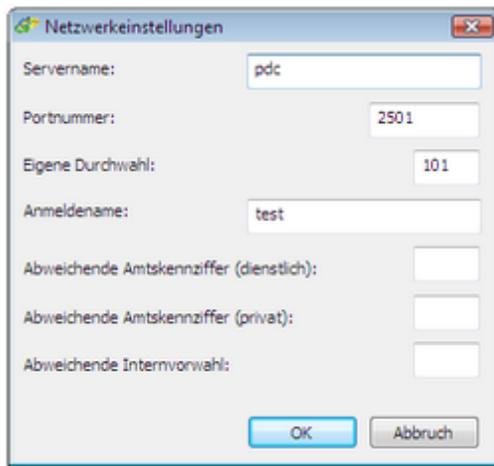
Aufgaben

Der CTI-Browser ist ein Testclient und kann als Entwicklungswerkzeug eingesetzt werden. Die Funktionen der Clientschnittstelle werden hier einzeln aufgerufen. Gleichzeitig wird der Datenfluß vom Server angezeigt. So ist es möglich Codesequenzen zu testen ohne direkt zu programmieren. Vor Einsatz des CTI-Browsers sollten Sie sich mit der [Schnittstelle](#) vertraut machen.



Bedienung

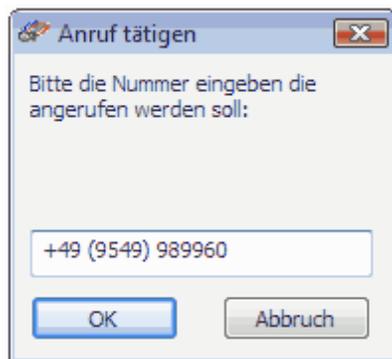
- Stellen Sie zuerst sicher dass der CTI-Server installiert ist und läuft.
- Der Client sollte auf dem Testplatz ebenfalls installiert sein. Beenden Sie den Client falls er aktiv sein sollte.
- Starten Sie dann den CTI-Browser.
- Auf der linken Seite sehen Sie nun die Befehle aus der [ActiveX Referenz](#). Die gleichnamigen C++ Funktionen haben in der Praxis noch einen Präfix "CTI_" vorangestellt. Eine Kurzbeschreibung des ausgewählten Befehls finden Sie in der Statuszeile. Die Befehle lassen sich mit einem Doppelklick starten. Einige Befehle werden erst angezeigt wenn die Verbindung zu Server steht.
- Beginnen Sie mit [NNetworkConfigDialog](#). In dem folgenden Dialog sollten Servername, Portnummer und eigene Durchwahl richtig eingetragen sein.



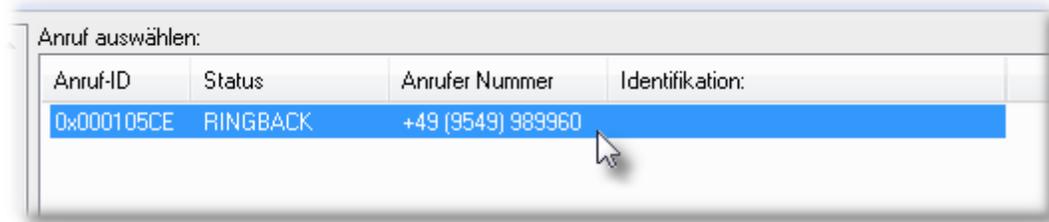
- Starten Sie als nächsten Befehl [NStartClient](#). Ein Dialog mit den Verbindungsoptionen wird daraufhin eingeblendet. Wählen Sie die Optionen wie unten gezeigt aus. Bestätigen Sie mit OK.



- Die Verbindung zum CTI-Server wird jetzt hergestellt. Danach erscheinen in der linken Liste wesentlich mehr Befehle. Starten Sie jetzt TMakeCall. In den folgenden kleinen Dialog können Sie eine Rufnummer eingeben und mit OK auswählen.



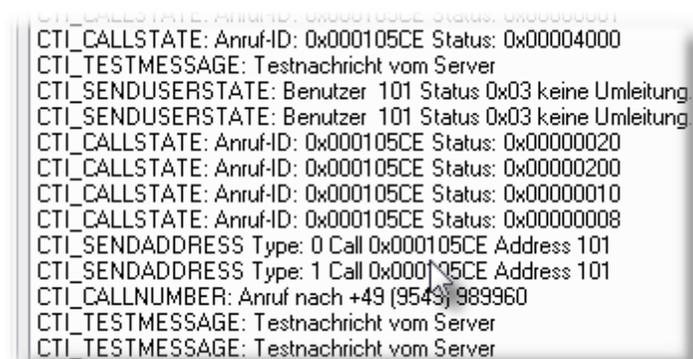
- Rechts oben im Browser wird der Zustand des Anrufs angezeigt sowie die Rufnummer, sofern diese ermittelt werden konnte.



Anruf auswählen:

Anruf-ID	Status	Anrufer Nummer	Identifikation:
0x000105CE	RINGBACK	+49 (9549) 989960	

- Rechts unten sind alle Events sichtbar die an den Client gesendet werden.



```
CTI_CALLSTATE: Anruf-ID: 0x000105CE Status: 0x00000001
CTI_CALLSTATE: Anruf-ID: 0x000105CE Status: 0x00004000
CTI_TESTMESSAGE: Testnachricht vom Server
CTI_SENDUSERSTATE: Benutzer 101 Status 0x03 keine Umleitung
CTI_SENDUSERSTATE: Benutzer 101 Status 0x03 keine Umleitung
CTI_CALLSTATE: Anruf-ID: 0x000105CE Status: 0x00000020
CTI_CALLSTATE: Anruf-ID: 0x000105CE Status: 0x00000200
CTI_CALLSTATE: Anruf-ID: 0x000105CE Status: 0x00000010
CTI_CALLSTATE: Anruf-ID: 0x000105CE Status: 0x00000008
CTI_SENDADDRESS Type: 0 Call 0x000105CE Address 101
CTI_SENDADDRESS Type: 1 Call 0x000105CE Address 101
CTI_CALLNUMBER: Anruf nach +49 (9549) 989960
CTI_TESTMESSAGE: Testnachricht vom Server
CTI_TESTMESSAGE: Testnachricht vom Server
```

- Mit [TCallDrop](#) können Sie das Gespräch beenden.
- Probieren Sie nach und nach weitere Befehle aus. Auf einigen Anlagen werden nicht alle Möglichkeiten vorhanden sein. Einen groben Überblick über Leistungsfähigkeit einer Anlage erhalten Sie mit dem [Leitungsbeobachter](#). Dieses Programm ist im Setup enthalten, Sie können die Datei auch [hier](#) herunterladen.

Siehe auch

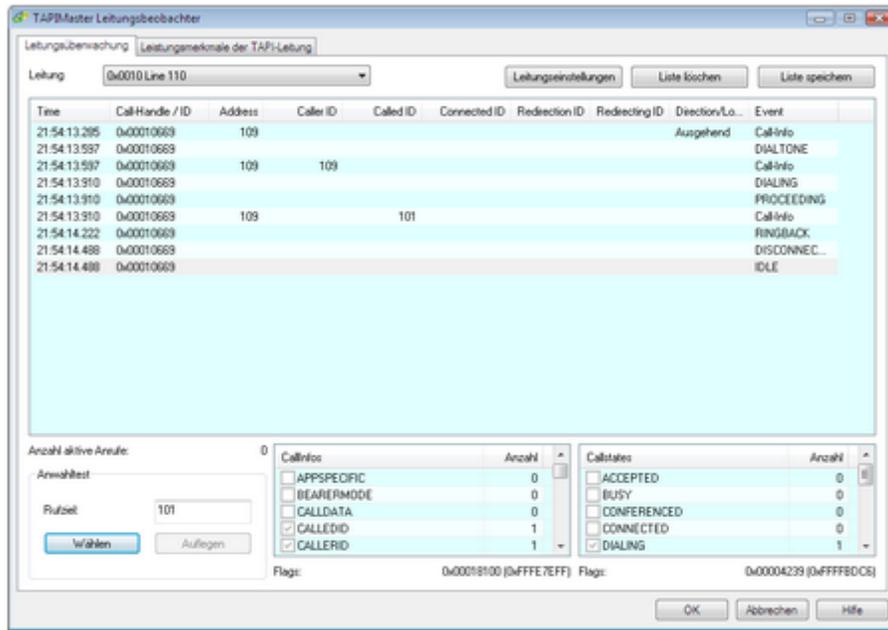
[Tools](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.5.2 Leitungsbeobachter

Der Leitungsbeobachter liegt in abgespeckter Form auch dem Installationspaket bei. Sie können mit dem Programm sehen, was auf einer TAPI-Leitung passiert. Sollte ein Client nicht mehr auf die Telefonanlage reagieren lohnt es sich den Leitungsbeobachter zu starten. Sie sehen dann ob z. B. Rufnummern übertragen werden oder die TAPI überhaupt noch arbeitet. Auf der Seite 2 des Programms sehen Sie die Leistungsmerkmale, welche der Hersteller verspricht. Die neueste Version des Programms können Sie [hier](#) herunterladen.



Siehe auch

[Tools](#)

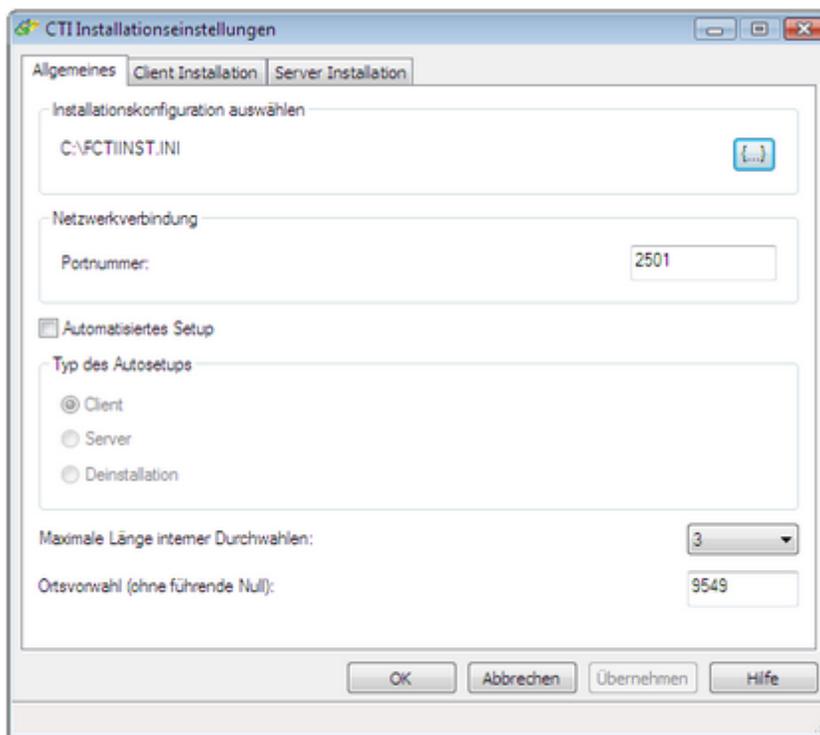
[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.5.3 Setup-Konfigurator

Wenn Sie Ihre eigene Software beim Kunden installieren, erfordert das in der Regel eine vorherige Installation der CTI-Software. Was beim Server noch erträglich ist kann bei mehreren hundert Clients nicht mehr manuell erfolgen. Deshalb haben Sie mit dem Setup-Konfigurator die Möglichkeit, automatisierte Setups für [Server](#) und [Client](#) zu erstellen, die dann im Hauptsetup eingebunden werden können.

- Auf der Seite "Allgemeines" finden Sie die Einstellungen welche Client und Server betreffen. Wählen sie zuerst die Konfigurationsdatei aus dem Setup-Verzeichnis aus.
- Die Portnummer ist für Client und Server gleich.
- Wenn das Setup gestartet wird, erscheint normalerweise ein Wizard der Sie durch die Installation führt. Das ist bei integrierten Clientsetups wenig sinnvoll. Markieren Sie "Automatisches Setup", um die Installation in einem Stück durchzuführen. Sie können dann ein Setup für Client, Server oder eine Deinstallation generieren. Alternativ dazu können Sie die SETUP.EXE mit dem Parameter -D (Deinstallation), -S (Server) oder -C (Client) starten, um eine automatische Installation zu erzwingen. Beim Client kann die Durchwahl mit angegeben werden, SETUP.EXE -C 251 installiert einen Client mit der Durchwahl 251.
- Die maximale Länge der internen Durchwahlen kann von 2 bis 4 Stellen haben.
- Die Ortsvorwahl wird ohne führende Null eingetragen.



[Benötigte Dateien](#)

[Client-Setup](#)

[Server-Setup](#)

Siehe auch

[Tools](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.5.3.1 Dateien

Je nach Setup sind verschiedene Dateien erforderlich. Kopieren Sie diese Dateien in ein Verzeichnis und fügen Sie sie Ihrem Setup hinzu. Die Datei FCTINST.INI sollte dann mit dem Setup-Konfigurator vorbereitet werden. Komprimieren Sie das Verzeichnis mit Winzip und lassen Sie die SETUP.EXE nach dem Entpacken aufrufen.

Dateiname	Server	Standard Client mit Outlook und Datenbank Telefonbuch	Standardclient ohne extra Telefonbücher	Fremdanwendung mit COM und Neustartmöglichkeit	Fremdanwendung ohne COM und Neustartmöglichkeit
FCTIACD.DLL	X				
FCTIACVW.EXE	X				
FCTICAN.DLL	X	X	X	X	X
FCTICAN.TLB	X			X	
FCTICLNT.DLL		X	X	X	X
FCTICLNT.EXE		X	X		
FCTICLNT.TLB				X	
FCTICLNT.TSP		X			
FCTIDATA.DLL	X				
FCTIDATA.TLB	X				
FCTIDBBK.DLL		X			
FCTIHELP.CHM	X	X	X		X
FCTIHOOK.DLL		X	X		
FCTIINI.DLL	X	X	X	X	X
FCTIINST.INI	X	X	X	X	X
FCTILANG.DLL	X	X	X	X	X

FCTIOUT.DLL		X			
FCTIPHHK.DLL		X	X	X	X
FCTIREST.EXE		X	X	X	
FCTIRMVE.EXE	X	X	X	X	X
FCTISCUT.EXE		X	X		
FCTISRV.CPL	X				
FCTISRV.EXE	X				
LW.EXE	X				
SETUP.EXE	X	X	X	X	X
STOPTAPI.EXE	X				

Siehe auch

[Setup-Konfigurator](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.5.3.2 Client-Setup

Verbindungsoptionen

Hier können die Optionen wie unter [Verbindungsoptionen](#) beschrieben einzeln aktiviert werden. Ein Aufruf von [CTI_OSetOptions](#) bzw. OOptions entfällt. Die Client-Installation kann komplett automatisiert werden. Beim ersten Start ist es erforderlich dass der Benutzer seine Durchwahl angibt, es sei denn die Installation wurde mit SETUP.EXE -C <Durchwahl> aufgerufen, z. B: SETUP.EXE -C 255.

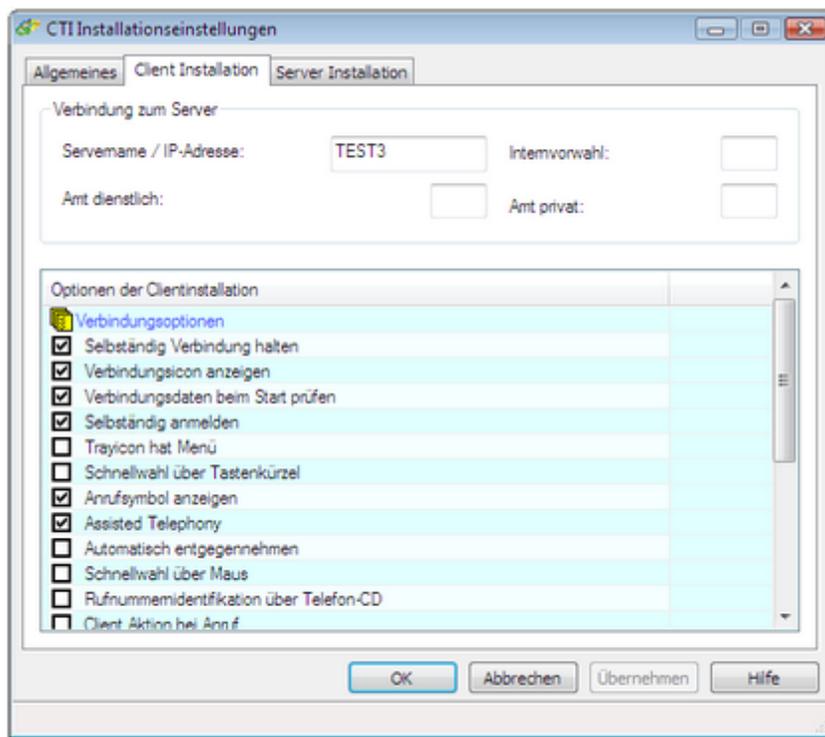
Installierte Komponenten

Normalerweise wird der Standardclient mit seinen Telefonbüchern installiert. Wenn Sie die Schnittstelle in Ihren eigenen Produkten einbauen ist das natürlich überflüssig. Wenn Sie ihn nutzen gibt es möglicherweise kein Outlook in der Firma. Sie können hier deaktivieren was Sie nicht benötigen.

Sonstige Einstellungen

Wird die [ActiveX](#) Schnittstelle nicht benutzt, kann auch auf die Registrierung der Schnittstelle verzichtet werden. Die Shortcuts auf dem Desktop lassen sich unterdrücken.

Das Setup überschreibt keine vorhandenen Durchwahlen, kann also auch zur Aktualisierung der Software eingesetzt werden.



Siehe auch

[Setup-Konfigurator](#)

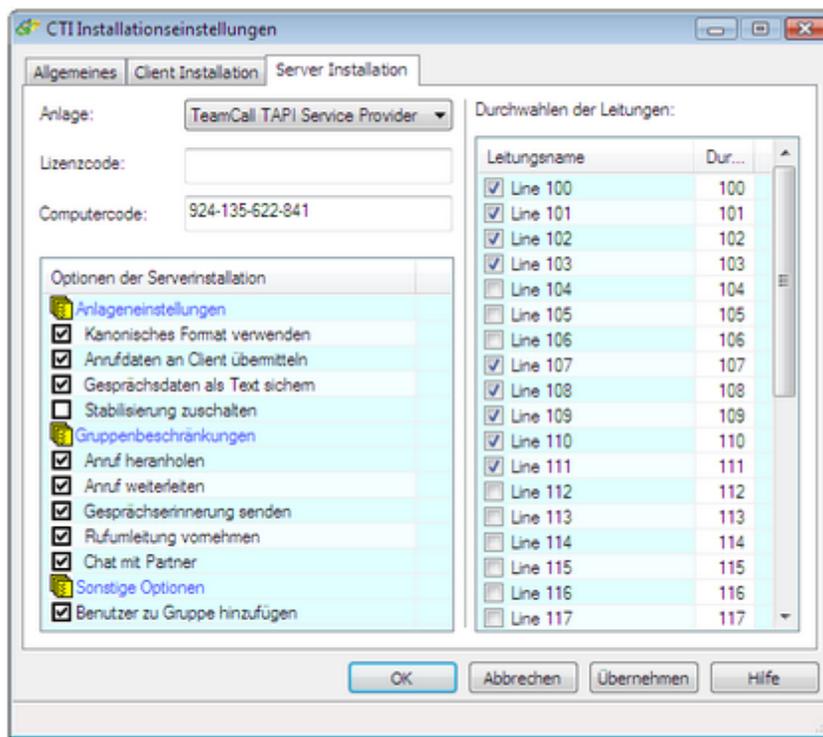
[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.5.3.3 Server-Setup

Allgemeines

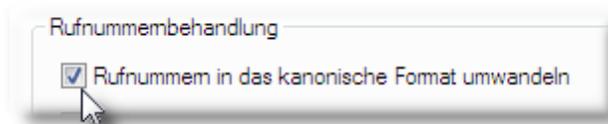
Wenn Sie eine [Installationskonfiguration](#) öffnen die der Kunde mit dem [Leitungsbeobachter](#) erstellt hat dann sind die Felder "Anlage", "Computercode" und "Durchwahlen der Leitungen" ausgefüllt. Wählen Sie zuerst die richtige Anlage aus. Aktivieren Sie die Durchwahlen welche verwendet werden sollen. Wenn bekannt kann der Lizenzcode eingegeben werden.



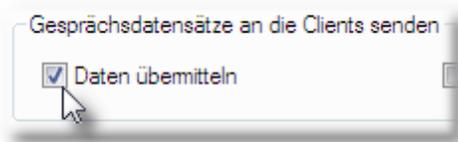
Anlageneinstellungen

Diese Optionen werden bei jeder Installation überschrieben.

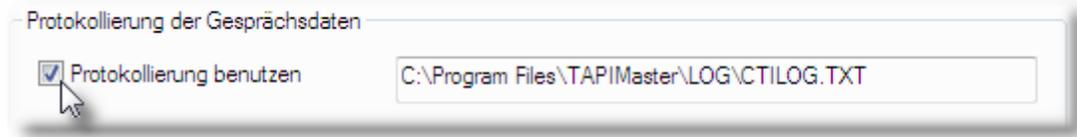
Es ist sinnvoll, Rufnummern in das kanonische Format zu formatieren, es sei denn Sie formatieren in Ihrem Client die Nummern nach Ihren eigenen Bedürfnissen. auf der Seite "Anlageneinstellungen" der Serverkonsole kann die Option jederzeit korrigiert werden.



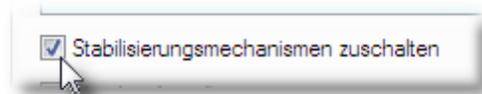
Wird eine Anruferliste auf dem Client geführt (z. B. Standardclient), so sollten die Gesprächsdaten an den Client übertragen werden. Option auch auf Seite "Datenbankschnittstelle" der Serverkonsole.



Wenn die Datenbankschnittstelle nicht benutzt wird werden die Gespräche normalerweise nicht mitprotokolliert. Sie können die Daten jedoch in einer Textdatei speichern lassen, um hier eine minimale Protokollierung zu haben. Diese lässt sich dann leicht in Excel einlesen und weiter auswerten. Option auch auf Seite "Datenbankschnittstelle" der Serverkonsole.



Sollten insbesondere bei großen Anlagen die Clients nicht mehr reagieren, z. B. keine Gespräche mehr aufgelegt werden können, so schalten Sie probeweise die Stabilisierung zu. Allerdings werden hier in einigen Fällen Gespräche mit von der TAPI falsch gelieferten [Gesprächszuständen](#) aufgelegt. Option auch auf Seite "TAPI-Sonderbehandlung" der Serverkonsole.

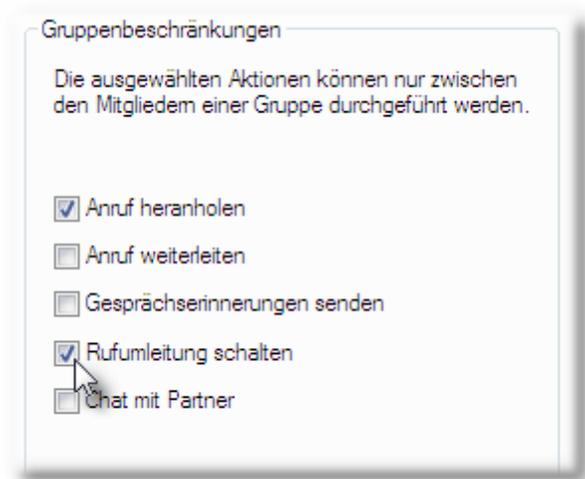


Gruppenbeschränkungen

Die Optionen werden ebenfalls immer überschrieben. Sie schränken folgende Funktionen ein:

[CTI_TPickup / TPickup](#)
[CTI_T_CallRedirect / TCallRedirect](#)
[CTI_TSendCallReminder / TSendCallReminder](#)
[CTI_TSetForwarding / TSetForwarding](#)
[CTI_EChatStart / EChatStart](#)

Sie finden diese Einstellungen dann auch auf der Seite "Gruppenverwaltung" der Serverkonsole.



Sonstige Optionen

Die vorhandenen Durchwahlen können alle in die erste Gruppe übernommen werden. Dann funktioniert die Anzeige der Gruppenmitglieder und deren Zustände sofort. Aktivieren Sie diese Option nicht wenn Sie mehrere hundert Clients haben, da hier die Netzlast zu stark ansteigt.

Siehe auch

[Setup-Konfigurator](#)

[Send feedback to TAPIMaster®](#)
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.6 Kundenspezifische Anpassungen

Als Softwarehersteller oder Großhändler können Sie TAPIMaster® optisch an Ihre persönlichen Bedürfnisse anpassen lassen. Hinsichtlich der Namen, Texte, farbliche Gestaltung und Logos sind derzeit die unten aufgeführten Änderungen möglich.

[Installation](#)

[Client Schnittstelle](#)

[CTI Server](#)

[Serverkonsole](#)

[Leitungsbeobachter](#)

[Sprachen](#)

[Benötigte Ressourcen](#)

Siehe auch

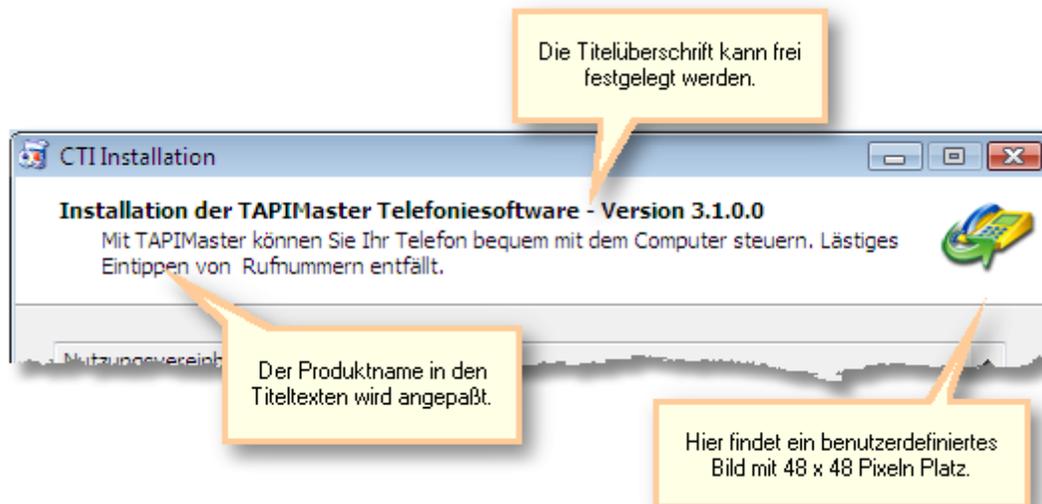
[Inhalt](#)

[Send feedback to TAPIMaster®](#)

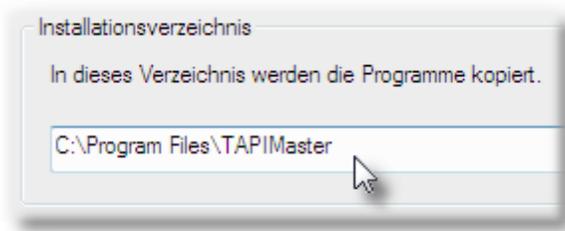
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.6.1 Installation

Beim Installationsdialog können Überschriften, Produktname und Logo geändert werden.



Das Installationsverzeichnis entspricht dem Produktnamen.



Siehe auch

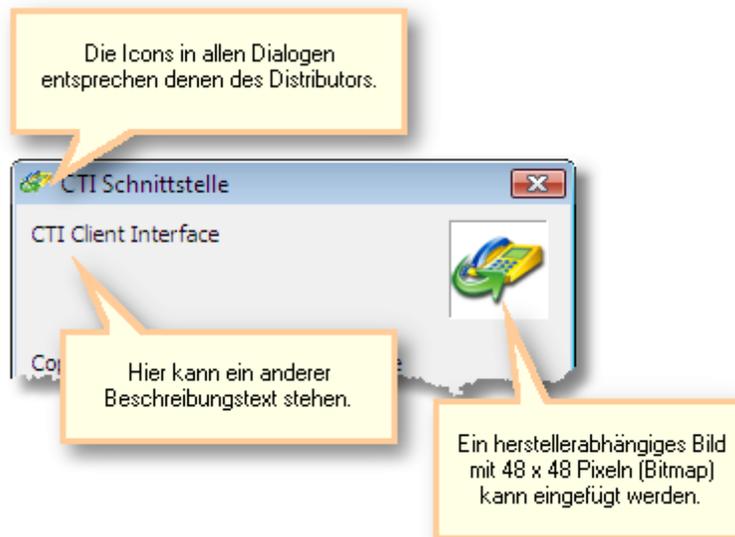
[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.6.2 Client Schnittstelle

Die Clientschnittstelle verfügt über einen Infodialog, welcher mit [EShowInfoDialog](#) bzw. [CTI_EShowInfoDialog](#) angezeigt werden kann.



Zur Laufzeit kann das Verbindungssymbol in der Taskleiste mit der Funktion [OConnectionIconOptions](#) / [CTI_OConnectionIconOptions](#) geändert werden.

Siehe auch

[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

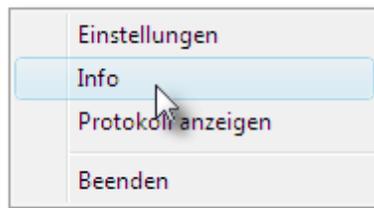
1.6.3 CTI Server

Der Server platziert ein Icon in die Taskleiste wenn er läuft. Es gibt ein Icon für den aktiven Server und eins für den Fall dass der Server Probleme beim Start hatte oder angehalten wurde.

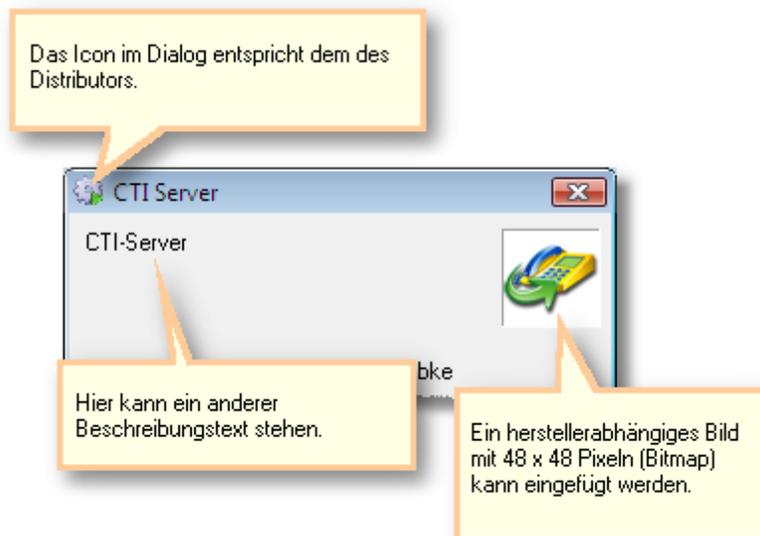


Über dem Icon wird ein benutzerdefinierter Tooltip angezeigt.

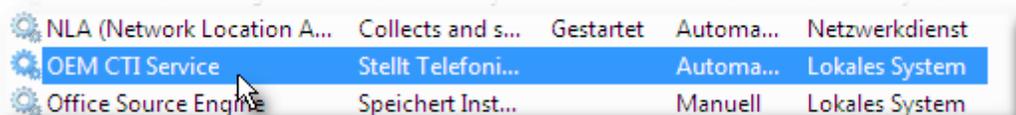
Über das Servermenü lässt sich ein Informationsdialog anzeigen.



Im Dialog lassen sich wieder einige Elemente ändern.



Der CTI-Server läuft normalerweise als Windows-Dienst. Dort kann er unter einem anderen Namen erscheinen.



Siehe auch

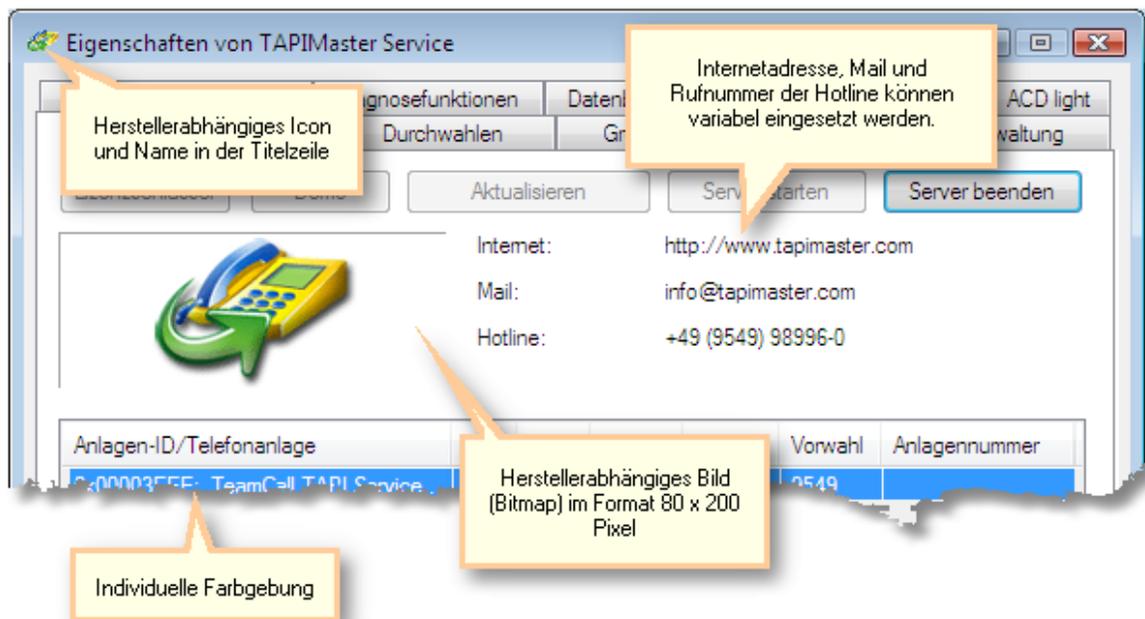
[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

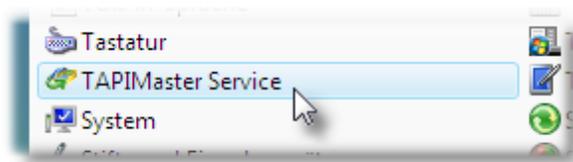
© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.6.4 Serverkonsole

Texte, Icons, Bilder und Farbgebung können angepaßt werden.



Die Serverkonsole ist über die Systemsteuerung erreichbar. Der dort angezeigte Name ist austauschbar.



Siehe auch

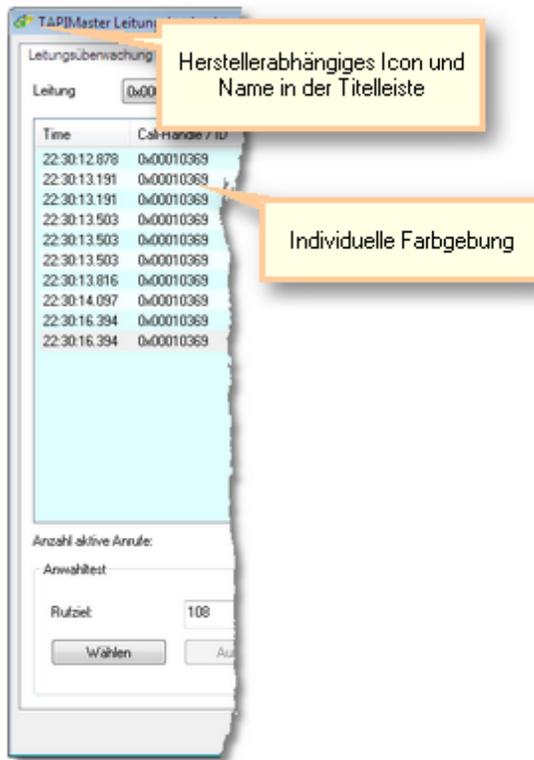
[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.6.5 Leitungsbeobachter

Der Leitungsbeobachter kann ebenfalls an die Software des Herstellers angepaßt werden.



Siehe auch

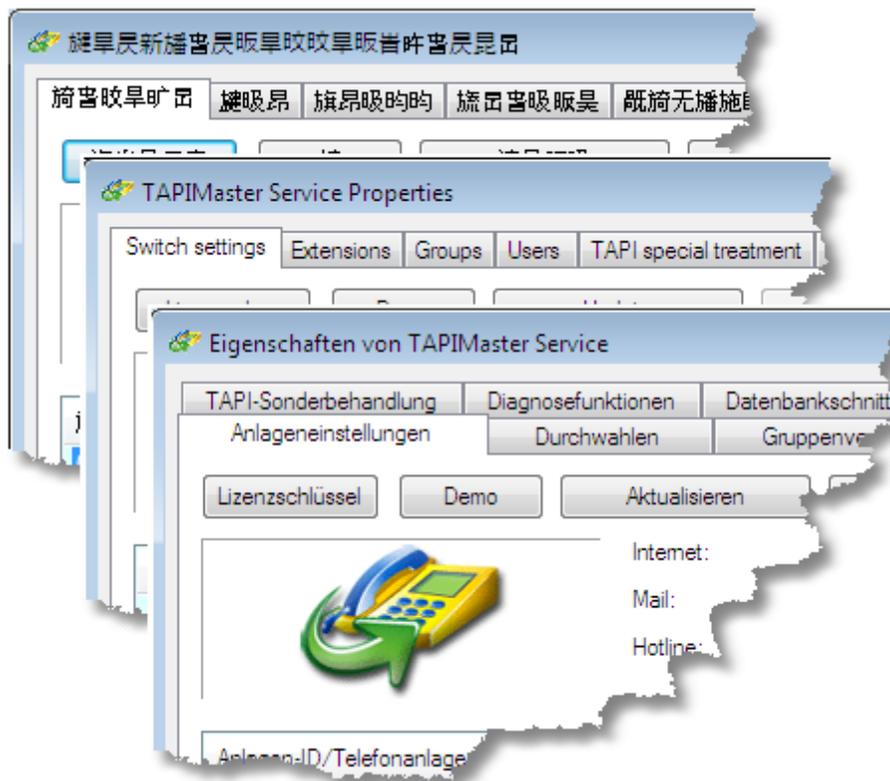
[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.6.6 Sprachen

TAPIMaster® holt sich seine Sprachressourcen aus der FCTILANG.DLL. Wenn Sie diese Datei mit einem Ressourceneditor übersetzen, können Sie TAPIMaster® in anderen Sprachen nutzen. Für Chinesisch, Japanisch und einige andere Sprachen fordern Sie bitte die Unicode-Version von TAPIMaster® an.



Wichtig bei der Übersetzung der Sprachressourcen: Entfernen Sie keine Formatierungszeichen wie %d, %s usw. Verändern Sie diese Anweisungen auch nicht in ihrer Reihenfolge. Entfernen oder verändern Sie auch keine Formatanweisungen wie \n, \r, \t. Programmabstürze können die Folge sein, für die richtige Funktion kann dann keine Garantie mehr übernommen werden.

```

639 ... Texte in Anführungszeichen
640 Trennzeichen:
641 (Kom von %s) %s
642 Sekunden
643 Leitung zurücksetzen

```

Siehe auch

[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

1.6.7 Benötigte Ressourcen

Zusammenfassung der benötigten Bilder, Texte und Konstanten

Um TAPIMaster® an das Erscheinungsbild eines Softwareherstellers bzw. Distributors anzupassen sind folgende Ressourcen erforderlich:

Bitmaps

Quadratisches Bitmap 48 x 48 Pixel.

Beispiel



Rechteckiges Bitmap 200 x 80 Pixel

Beispiel



Icons

Je ein Icon für die Anwendungen, den aktiven und den inaktiven Server.

Texte

- Produktname
- Firmenname
- Servicename, wird in Systemsteuerung und Dienstverwaltung angezeigt
- Internet-Adresse
- Mail
- Rufnummer der Hotline
- Kurzbeschreibung der [Client-Schnittstelle](#) und des [Servers](#)
- Überschrift im [Installationsdialog](#)

Farbkonstanten

Die Farbkonstanten werden als RGB-Werte angegeben z. B. RGB(192,192,192) für hellgrau.

- Textfarbe in Listen, normal
- Textfarbe in Listen, Teilüberschriften
- Textfarbe in Listen, deaktivierter Text
- Texthintergrund in Listen, veränderliche Daten
- Texthintergrund in Listen, Bereich außerhalb der Einträge
- Texthintergrund in Listen, gerade Zeilen
- Texthintergrund in Listen, ungerade Zeilen

Siehe auch

[Kundenspezifische Anpassungen](#)

[Send feedback to TAPIMaster®](#)

© 2020 Tino Kasubke. Alle Rechte vorbehalten

Index

- A -

- Abmelden vom CTI-Server 116, 179
- Anlagenrufnummer 165, 235
- Anmelden am CTI-Server 116, 122, 166, 178, 236, 251
- Anruf 164, 234
 - Adreßübermittlung 169, 238
 - Auflegen 134, 197
 - Entgegennehmen 133, 196
 - Gespräch heranziehen 138, 202
 - Gesprächserinnerung senden 140, 157, 204, 228
 - Halten 134, 197
 - Halten beenden 136, 200
 - Identifikation im Server 323, 329, 342, 347
 - Makeln 135, 198
 - Rufnummernidentifikation 164, 234
 - Rufnummernübermittlung 156, 228
 - Telefon-CD 164, 234
 - Wählen 138, 202
 - Wählen über Proxytelefon 139, 203
 - Wahlwiederholung 139, 203
 - Weiterleiten 134, 198
 - Zwei Anrufe verbinden 136, 199
- Anruferliste anzeigen 171, 222
- Anrufsymbol
 - Anzeige bei eingehenden Anrufen 128, 252
 - Entfernen 140, 204
- Anwahl von markiertem Text 126, 252, 253
- Assisted Telephony 121, 250
- Automatisierte Setups 366

- B -

- Befehlsreferenz
 - Client 4
 - Server 264
- Benutzerdefinierte Nachrichten 154, 161, 218, 232, 332, 349
- Benutzernachrichten 152, 173, 216, 240

- C -

- Chat
 - Sitzung beenden 147, 159, 212, 230
 - Sitzung eröffnen 148, 159, 212, 231
 - Text übertragen 148, 158, 212, 230
- Client - Neustart 118, 160, 180, 231
- Clients ohne Installation verwenden 355
- CTI-Browser 362

- D -

- Dateien für Installation 367
- Datenbankabfragen 245
- Datenstrukturen 244
- Deinstallation 359

- E -

- Einbettung in Setups 355
- Eingebettetes Client-Setup 356
- Eingebettetes Server-Setup 357
- Eingehende Gespräche
 - Automatisch entgegennehmen 121, 251
 - Entgegennehmen 133, 196
 - Reaktion 130, 189, 192
- Events
 - (ActiveX) 155
 - (C++ CTI API) 220

- F -

- Funktionen
 - Client 4
 - Server 264

- G -

- Gesprächsdatensätze 167, 237, 331, 349
- Gesprächserinnerung senden 140, 157, 204, 228
- Gesprächszustände
 - Auflistung 158, 249
 - Übertragung 229
- Gruppenmitglieder
 - Auflistung 163, 233

Gruppenmitglieder
Status 170, 238

- I -

Info-Dialog 153, 218
Inhalt 1
Inhaltsverzeichnis 1
Installationskonfiguration 358

- K -

Konferenz
Anruf hinzufügen 132, 195
Status anzeigen 160, 232
Teilnehmer anrufen und hinzufügen 133, 196
Konfigurationsdatei
Client 126, 187
Setup 366
Konstanten 248

- L -

Ländereinstellungen 165, 235
Leitungsbeobachter 365
Lizenzen 2
Lizenzinformationen 165, 235

- M -

Maßgeschneidertes Server-Setup 357
Menü 129, 166, 221, 254

- N -

Nachrichten senden 152, 173, 216, 240
Nachrichten vom Administrator 170, 239
Netzwerkdialog 117, 179
Netzwerkfunktionen
(ActiveX) 115
(C++ CTI API) 175
Netzwerknachrichten 226
Netzwerkverbindung 118, 119
Parameter auslesen 118, 119, 177, 189
Parameter einstellen 117, 118, 119, 128, 179,
181, 182, 192

Neustart vorbereiten 118, 160, 180, 231

- O -

Optionen
(ActiveX) 120
(C++ CTI API) 184

- P -

Programmierbeispiele
Anwahl über IP-Schnittstelle 352
Client 256
CTI-Client mit C++ 259
CTI-Client mit Delphi 260
Excel als CTI-Client 258
Server 351
Server-Schnittstelle in Excel / VB 354
SQL-Handling 261
Programmierhilfsmittel 361
Programmertools
CTI-Browser 362
Setup-Konfigurator 366

- R -

Rufumleitung 137
Abfragen 137, 200, 201
Info über fremde Umleitung 167, 236
Übermittlung beim Anmelden 163, 233
Umleitung schalten 137, 141, 205

- S -

Schnittstellen
Client 3
Server 263
Testen von Befehlen 362
Serverschnittstelle
Anrufen 321
Gespräch heranholen 321
Gespräch weiterleiten 322
Initialisierung 320, 339
Konferenz 312
Makeln 311
Verbinden 325
Setup 355

Setup-Konfigurator 366
 Setups
 Automatisierte Setups 366
 Client 368
 Dateien 367
 Installationskonfiguration 358
 Server 369
 SQL
 Abfrage beenden 143, 172, 207, 242
 Abfrage fortsetzen 145, 209
 Abfrage starten 143, 207
 Anmelden 145, 172, 209, 242
 Beispielprogramm 261
 Daten empfangen 171, 240
 Datentypen 245
 Funktionsübersicht (ActiveX) 142
 Funktionsübersicht (C++ API) 206
 Telefonbuchdatensätze abfragen 144, 162, 208, 241
 Strukturen 244

- T -

Tastenkürzel
 Anwahl 124, 126, 186, 189, 252
 Gespräch auflegen 125, 186, 190
 Telefoniefunktionen
 (ActiveX) 131
 (C++ CTI API) 194
 Testclient 362
 Tools 361

- U -

Unicode 165, 234

- V -

Verbindung 115
 automatisch halten 122, 251
 Eigenschaften 127, 188, 191
 Initialisieren 119, 182, 183
 Status 161, 223, 224, 239
 Status abfragen 115, 176
 unterbrechen 115, 177
 wieder herstellen 115, 176
 Verbindungsdaten prüfen 123, 252

Verbindungsdialog 117, 179
 Verbindungsicon
 In Taskbar 123, 129, 253
 Menü 129, 166, 221, 254
 Verbindungsoptionen 250
 Versionsanzeige 153, 218
 Verweise
 Client 113
 Server 308
 Vordergrund 168, 225
 Vorwahl 165, 235

- W -

Wahlwiederholung 139, 203

- Z -

Zusatzfunktionen
 (ActiveX) 147
 (C++ CTI API) 211

Endnotes 2... (after index)

Back Cover