

Produkt-Highlights

- ✓ Visual Basic, Delphi, Script u.v.m., statt C++
- ✓ Objects und Collections statt Handles
- ✓ Event-Handler statt Call-Back-Funktionen
- ✓ Objektmodell statt "native functioncalls"
- ✓ Eingehend und ausgehend telefonieren
- ✓ First- und Third-Party Support
- ✓ Windows COM-basierte Telefonie
- ✓ Keine Run-Time Lizenzen
- ✓ Windows 95, Windows 98, Windows NT4 und Windows 2000
- ✓ Einheitliche TAPI-Schnittstelle für alle 32-Bit Windows-Betriebssysteme
- ✓ Einheitliche Programmierschnittstelle für unterschiedliche TAPI-Telefonanlagen
- ✓ Einfachste Implementierung der Telefonie in jede COM-fähige Windows-Anwendung
- ✓ Caller-ID Identifikation
- ✓ Rufweiterleitung
- ✓ Rückfrage, Makeln und Konferenz
- ✓ Ausgehende Telefonate mit automatischem, *standortabhängigem und internationalem Least-Cost-Routing*
- ✓ Ausführliche Beispiele
- ✓ Ausführliches Debug-Protokoll über alle Windows TAPI-Aktivitäten
- ✓ Beliebige viele Devices gleichzeitig
- ✓ Beliebige viele gleichzeitig geöffnete Lines, mit jeweils mehreren aktiven Calls
- ✓ TELES - iPBX zertifiziert
- ✓ Siemens Hicom getestet

- OR Network
- Eiserne Hand 11
- 35305 Grünberg
- Fon 0 64 01 / 2 20 12-0
- Fax 0 64 01 / 2 20 12-99
- www.OR-Network.net

COM (ActiveX) basiertes TAPI für Windows 9x, NT4 und Win2000

CTI-Funktionalitäten sind heutzutage für Windows-Programme Pflicht; denn kaum ein Kunde möchte noch auf den Komfort einer Computer gestützten Telefonie verzichten.

```
Set actTAPI = WScript.CreateObject('actTAPI.TAPI')
actTAPI.Initialize('MyApp')
For Each d in actTAPI.Devices
    s = s & 'DeviceName = ' & d.Name & vbCr
Next
MsgBox s
```

Bedingt durch die proprietäre Windows-TAPI-Schnittstelle (Telephone Application Programming Interface) ist die Implementierung von CTI-Funktionalitäten mit modernen Programmier- und Scriptsprachen gar nicht oder nur mit hohem Aufwand möglich. In der Regel müssen Anwendungsprogrammierer auf C++ zurückgreifen, entsprechendes Know-How und genügend Man-Power für eine CTI-Implementierung zur Verfügung stellen.

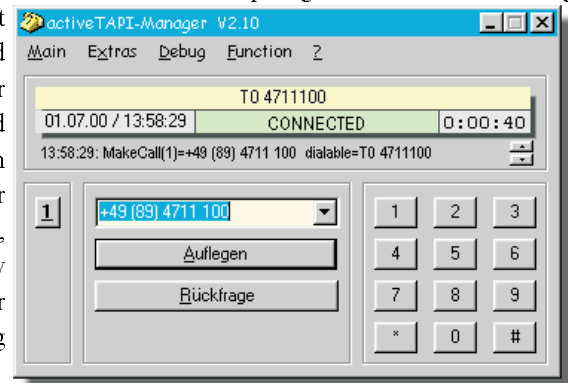
Die activeTAPI-COM-Objects

- ✓ **Erprobt und zertifiziert** sparen die activeTAPI-COM-Objects Zeit und Geld bei der Implementierung von CTI-Funktionalitäten in jede Windows-Anwendung.
- ✓ Die activeTAPI-COM-Objects machen Ihre Anwendung unabhängig von den vielen Besonderheiten (und teilweise auch Fehlern) der TSPI-Treiber der **verschiedenen Telefon- und Anlagenhersteller**.
- ✓ Mit den activeTAPI-COM-Objects steht Ihnen CTI unabhängig von der Programmiersprache zur Verfügung. Zum Beispiel in **Visual Basic**, in Visual Basic Script, in VBA, in **Delphi** und vielen mehr.

- ✓ Die activeTAPI-COM-Objects werden stetig erweitert und verbessert und mit weiteren Telefonen und Anlagen getestet. Sie profitieren unmittelbar davon, weil Sie Ihre Anwendung nicht mehr an neue TSPI-Treiber der Hersteller anpassen müssen.

Der activeTAPI-Manager

Der activeTAPI-Manager basiert auf den activeTAPI-COM-Objects und setzt deren Funktionalität vollständig um. Ursprünglich als Test-Anwendung



für die activeTAPI-COM-Objects gedacht, ist der activeTAPI-Manager seit der Version 2 eine vollständige activeTAPI-Applikation. Der active-

```
Set activeTAPI = CreateObject('actTAPI.TAPI')
Ret = activeTAPI.Initialize('MyApp')
If Ret = TAPIRESULT_SUCCESS Then
    Ret = activeTAPI.LineOpen('ESP Line 6')
    Ret = activeTAPI.MakeCall(DestAddress='4711')
```

TAPI-Manager ist von der eigentlichen CTI-Funktionalität völlig entkoppelt und beinhaltet nur die Funktionalitäten des User-Interface.

„Die zertifizierten activeTAPI-COM-Objects beschleunigen den Prozess der TAPI-Implementierung erheblich und reduzieren den Aufwand zur Fehlersuche auf ein Minimum.“