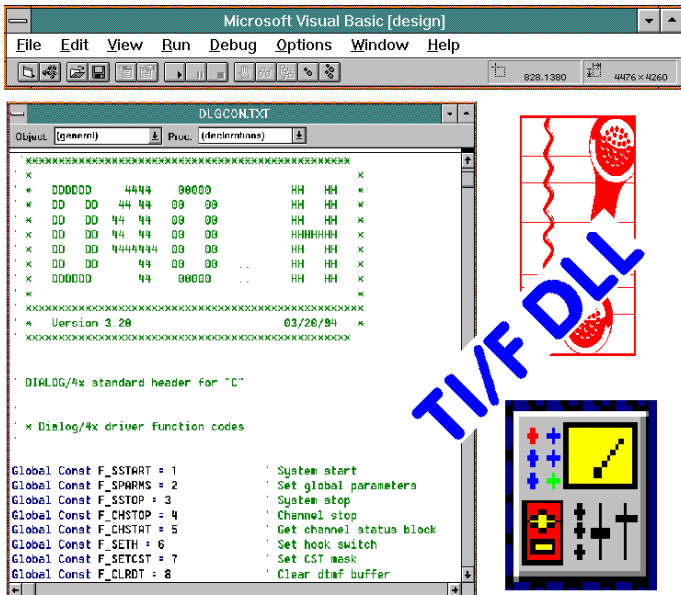


# Windows Voice Mail Now!

*TI/F DLL™: The Windows Voice Mail Driver*



## Make Your Job Easier!

Voice developers rave about *TI/F DLL™* because it makes their life easy! With it, developing voice apps with VB, C, and C++ is a breeze.

How? By letting you make standard driver function calls from your windows apps, letting you port your voice program in record time.

Add Voice Mail power to your Windows applications with our Telephone Interface (TI/F) Dynamic Link Library (DLL). Use C or C++ or Visual Basic to create applications for Windows without wanting to jump out of one. And use *TI/F DLL* to control your Dialogic (and compatible) voice communications hardware.

When we created the *TI/F DLL* program, we

were just trying to get our job done. Our *VFEdit®* Professional Prompt Editor no longer worked with Windows 3.0. The voice board manufacturers had left this gigantic hole. So we set about devising a way to plug the hole. Apparently everyone else with voice processing was having the same problem. Before we knew it, we were selling the “plug” — *TI/F DLL* — right and left.

And now, seven years later, the same hole is in newer versions of Windows, and *TI/F DLL* is more popular than ever. Without it, the Dialogic, Rhetorex, Bicom and NewVoice real mode drivers cannot interface with Windows programs. Because of this, *TI/F DLL* recently won *Computer Telephony* magazine's Product of the Year award.

## Proven Technology

Rather than reinvent the wheel, developers can avoid the headaches and use *TI/F DLL*. After years of perfecting this interface, it's definitely worth it. Most developers opt for buying *TI/F DLL* because of its solid track record.

And now with Windows '95, some users are choosing to avoid the proprietary (and expensive) drivers that lock developers to a particular manufacturer, and stick with the mature DOS real-mode driver technologies.

## Highlights:

- Use Dialogic compatible boards without expensive proprietary drivers.
- Swap popular Dialogic compatible boards without changing application software.
- Ideal for legacy applications requiring backward compatibility with existing driver solutions.
- Supports up to 64 lines per system.
- Extended support agreements and royalty-free licenses are available.

*“To the advanced programmer looking for the serious tool with all the teeth of a great white shark, TI/F DLL will provide you with all of the bite you can handle.”*

*-Robert Cristello, Computer Telephony Magazine.*



# T/F DLL

For a risk free test drive, download our "Working Demo". We think you'll like it!

## Extensions:

DskFilCls: Close file opened with DskFilOpn.  
DskFilCre: Creates and opens a file.  
DskFilLen: Return length of file.  
DskFilOpn: Open a file for playback and recording.  
DskFilPos: Move the file pointer.  
MemGetRea: Allocate DOS real memory.  
MemRelRea: Release DOS real memory.  
MVBCopMem: VB Copy protected memory.  
MVBCopStr: VB Copy protected mode memory string.  
MVBGetOff: VB Return pointer offset.  
MVBGetSeg: VB Return pointer segment.  
TIFDevGet: Request device type use.  
TIFDevRel: Release device type.  
TIFLinGet: Request line. 0 for next.  
TIFLinRel: Release use of a line.  
TIFPolEvt: Synchronize event queues.  
TIFPrtGet: Request multiplexer port.  
TIFPrtRel: Release multiplexer port.  
TIFSupIni: Initialize TI/F DLL use.  
TIFSupTrm: Terminate TI/F DLL use.  
TIFSupVer: Get TI/F DLL version #.

## Replacement Functions:

calld40x: Use CalDxxInt() for I/F.  
getevt: Use GetLinEvt() or GetDevEvt() to retrieve event.  
gtevtblk: Use GetLinEvt() or GetDevEvt() for event block.  
startsys: Initialize hardware.  
startamx: Initialize multiplexer.  
stopsys: Stop hardware system.  
stopamx: Stop multiplexer system.

## Standard Functions (partial listing):

amx\_msk: Control port signals & events.  
amx\_off: Break all switch connections.  
calld40: Low-level driver interface.  
callp: Call number, monitor progress.  
clrcpb: Zero Channel Parameter Block.  
clrdcb: Zero Dialog Control Block.  
clrdtmf: Clear all digits from buffer.  
clrrwb: Zero Read Write Block.  
clrxrwb: Zero Extended RWB.  
d4xerr: Get ASCII err code message.  
dial: Dial an ASCIIZ string.  
dl\_addspddig: Add change speed digit  
dl\_addtone: Add a channel global tone.  
dl\_addvoldig: Add change vol digit.  
dl\_adjsv: Modify play speed/ volume.  
dl\_blddt: Build a global dual tone.  
dl\_blddtcad: Build cadence dual tone.  
dl\_bldst: Build global single tone.  
dl\_bldstcad: Build tone with cadence.  
dl\_bldtngen: Build tone gen template.  
dl\_chgdur: Set call anal tone length.  
dl\_chgfreq: Set call analysis freq.  
dl\_chgqualid: Set qualify template.  
dl\_chgrepcnt: Set call analysis tone cadence count.  
dl\_clrsvcond: Clear speed/volume adjust

dl\_deltone: Delete chan global tones.  
dl\_distone: Disable chan tone detect.  
dl\_enthone: Enable chan tone detect.  
dl\_flushn: Flush tone info buffer.  
dl\_getcursv: Get speed/ volume setting.  
dl\_getextqual: Get qual template(ex).  
dl\_getqual: Read qual template.  
dl\_getsvmt: Get speed/ volume table.  
dl\_gettinfo: Read tone information.  
dl\_gtsernum: Get board serial number.  
dl\_initcallp: Initialize call analysis.  
dl\_playtone: Play a tone.  
dl\_selqual: Select qual template.  
dl\_setextqual: Set qual template(ex).  
dl\_setgtdamp: Global Tone Detect amp.  
dl\_setqual: Update qual template.  
dl\_setsvcond: Set speed/ volume adjustment conditions.  
dl\_setsvmt: Set speed/ volume table.  
getcar: Get call analysis results.  
getcm: Get shared communication area.  
getcstat: Get channel status.  
getdtmfs: Get DTMF digit string.  
getvctr: Get software interrupt vector.  
getver: Get driver version number.  
isdrvact: Is driver installed?  
playbuf: Play data from a buffer.  
playfile: Play data from a file.  
playuser: Play from the user buffer.  
putcm: Put shared communication area.  
putevt: Put event onto event queue.  
readdtmf: Get next digit in buffer.  
recbuf: Record data to a buffer.  
recfile: Record data to a file.  
recuser: Play from the users buffer.  
ring: Start ring cycle for port.  
sb\_route: Assign channel timeslot.  
sb\_rtrcvxmt: Assign independent timeslot to a channel  
sched: Give time slice to scheduler.  
setcparm: Set channel parameters.  
setcst: Set Call Status Transition.  
setdmask: Set Digit Control Mask.  
sethook: Set channel hook status.  
setiparm: Set channel parameters  
setxparm: Set ext global params.  
set\_ring: Set ring pattern for port.  
stopch: Stop multitasking function.  
sw\_off: Disconnect an AMX/8x switch.  
sw\_on: Connect switch on AMX/8x.  
wink: Set channel wink parameter.  
xcallp: Call a number and monitor call progress termination.  
xplayf: Play a file (extended).

## System Requirements

An IBM PC or compatible with a hard drive, DOS 3.3 or higher, and at least 512K of memory. A 486 DX, 640K and Windows 3.1x or higher is recommended.

## Pricing & Availability

- **T/F DLL** \$195
- Additional System licenses Call

## Raising Your Voice to New Levels

Voice Information Systems, Inc. improves your digital voice communications using signal processing techniques. VISI products include software for editing and managing digital audio and programming tools that help run your voice processing system. **T/F DLL** resulted from the need to create cooperative voice response applications for the Microsoft Windows environment.

Founded in 1986, VISI has been involved in voice processing since its inception. VISI's solid track record and superb reputation are a testament to the company's commitment to excellence.

We also offer consultation, custom programming, training and support services to companies working with voice and telephony. We've helped many companies in the industry acquire an edge with



our training and support. Let us do the same for you.

Sales: 800-234-VISI / 310-392-8780  
Fax: 800-234-FXIT / 310-392-5511  
BBS: 310-392-6610 (14.4-N-8-1)  
Web: [www.vinfo.com](http://www.vinfo.com)  
email: [sales@vinfo.com](mailto:sales@vinfo.com)