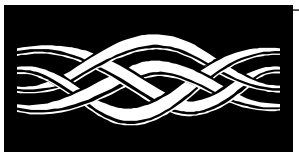
Microsoft®

Windows NT® Server

Server Operating System



White Paper

The Microsoft® Windows® Telephony Platform: Using TAPI 2.0 and Windows to Create the Next Generation of Computer-Telephony Integration

Microsoft Corporation ▪ One Microsoft Way ▪ Redmond, WA 98052-6399 ▪ USA

1196 Part No. 098-66258

Microsoft® BackOffice™

Microsoft® Windows NT® Server

TAPI 2.0

The Microsoft® Windows® Telephony API (TAPI) 2.0, which ships as part of Microsoft Windows NT® Server version 4.0 and Windows NT Workstation version 4.0 operating system, provides the most powerful and flexible platform for developing and using computer telephony applications. TAPI 2.0 abstracts the hardware layer, providing developers and users with the freedom of network and device independence. It is the only platform that enables applications for use on PSTN, ISDN, PBX, and IP networks.

Part of the Windows Open Systems Architecture (WOSA), TAPI 2.0 is supported by a full range of complementary APIs to enable a broad range of powerful and easy-to-use telephony and communications applications for a wide range of customers. In addition, new Microsoft ActiveX™ Controls further simplify and accelerate the process of development for a broader range of developers, enabling more customized telephony and related communications solutions.

Because TAPI 2.0 is built into Windows, it has a lower cost of deployment and ownership, and developers can create applications knowing that the large installed base of Windows 95 and Windows NT operating systems are already equipped to support their telephony applications. All of this makes Windows an excellent telephony and network communications platform for the future.

CONTENTS

INTRODUCTION

The Microsoft® Windows® Telephony API (TAPI) 2.0, which ships as part the Microsoft Windows NT® Server version 4.0 and Windows NT Workstation version 4.0 operating systems, provides the most powerful and flexible platform for developing and using computer-telephony integration (CTI) applications.

TAPI 2.0, which will also be released soon for Windows 95, abstracts the hardware layer, providing developers and users with the freedom of network and device independence. It is the only platform that enables applications for use on PSTN, ISDN, PBX, and IP networks.

The Windows telephony platform is significant because the telecommunications and computer industries are technologically converging. Both developers and end users will benefit from this convergence of the power of telecommunications with the friendliness, flexibility, and scalability of personal computers. The introduction of PC economics and business models into telecommunications is increasing choice, competition, and innovation while maintaining the high reliability people expect from their phone systems.

Computer-telephony integration creates huge opportunities for application developers, telephony network, switch, and hardware vendors, and end users. However, success is dependent upon CTI having at its foundation a robust, reliable operating system that is supported by a rich and powerful set of application programming interfaces (APIs) and related tools to enable this value-added chain to develop and flourish.

The Windows telephony platform provides developers, vendors, and end users with the best combination of solid operating system performance and tightly integrated APIs. The open standards of the Windows platform and TAPI 2.0 provide a rich foundation upon which the CTI industry can build. Complementing TAPI is a powerful collection of other Windows APIs and Win32® functions including the Messaging API (MAPI), the Speech API (SAPI), and Communication, Wave Audio, and Media Control Interface (MCI) functions.

Additionally, Microsoft ActiveX™ Controls give a much broader population of developers powerful and easy-to-implement "plug-in" software components that free developers from having to code the intricate plumbing of an application. Instead, ActiveX Controls allow the developers to use graphical development tools and concentrate on creating applications that are intuitive to use and deliver real user benefits

The abstraction of the hardware layer made possible with TAPI 2.0 is especially significant for CTI developers because historically the field has been a bewildering collection of proprietary switches, meaning that an application could only be written for one part of the industry. TAPI 2.0 frees developers from this burden, allowing a single application to be written that will work across switches.

Because TAPI 2.0 is built into Windows, it has a lower cost of deployment and ownership, and developers can create applications knowing that the large installed base of Windows 95 and Windows NT operating systems are already equipped to support their telephony applications. All of this makes Windows an excellent telephony and network communications platform for the future.

A CLOSER LOOK AT THE WINDOWS TELEPHONY PLATFORM

The Windows telephony platform has been designed with hardware abstraction and network independence to provide an open and flexible standard upon which the telephony industry can thrive.

The Challenge: Bringing Unity and Integration to a Segmented Landscape

Although CTI APIs have existed in various forms for more than two decades, early applications were often specific to closed, proprietary applications or platforms, and they lacked a compatibility that would allow horizontal integration with other communications applications.

Despite sitting next to one another on tens of millions of desktops, the telephone and computer have remained very poorly integrated. Analog modems are widely used, but do not support voice or many network capabilities, though that has changed with the development of UniModem V and related technologies. Still, analog modems are typically incompatible with the digital PBX telephone systems found in most offices. Other types of telephone networks abound, many of which are equally incompatible with today's modems, including Integrated Services Digital Network (ISDN), cellular, Centrex (a popular "virtual" PBX service offered by phone companies), and key systems (small PBXs), amongst others.

Microsoft provides a unifying platform for these two most common business technologies—the telephone and the personal computer. The Windows Telephony API helps bridge the gap between the telephone and computer by helping the PC to "understand" how telephone networks operate, and by letting programmers exploit network capabilities from within regular Windows-based applications.

Microsoft has been working with developers and vendors throughout the computer and telecommunications industries to create open APIs upon which all can build. The TAPI specification, introduced in 1993, was developed in conjunction with over 40 companies, including major chip and computer manufacturers, telecommunications equipment vendors, software developers, and network operators from around the world. Since then, the number of organizations contributing to the definition of TAPI's evolution has grown significantly making TAPI truly an industry-defined API. TAPI 2.0 represents a significant advancement of the API.

Supporting a Broad Range of Applications

To meet the needs of a wide range of users, a CTI API must support a broad range of applications. These include both "telephony-centric" applications, which have the management of telephone calls as their primary focus (from the simplest dialer or data communications application to complex call center management applications), and "telephony-enabled" applications—those which allow access to telephony features but do not have telephony as their primary focus. Examples of a "telephony-enabled" application would be a personal information manager or database management toolkit with rudimentary ability to place and track calls, most likely by leveraging the user's preferred telephony-centric ("call manager") application.

Some CTI architectures are focused on one facet of the problem, such as call centers (automatic call distribution, predictive dialing, and so on) or ISDN calls from personal computers. Widespread success of CTI requires a broad range of com-

mercial applications—and those applications will never be produced unless developers are convinced that drivers will be available for a wide range of underlying hardware. If the only drivers available are for large PBXs with ACD and predictive dialing features, or for specific ISDN interface boards, major software developers will be reluctant to make the investment to produce applications for that API.

The Windows Telephony API focuses on the "desktop"—generally, a single PC and telephone, however, the capability of performing "third-party" call control (on behalf of other desktops) is preserved where needed. The desktop is assumed, in the parameter structure of API functions, to be one endpoint of each call, thereby simplifying the API and making it easier for programmers to learn and use. The underlying protocol used to control the telephone network or PBX can be either first-party or third-party.

Bringing CTI to Common Hardware Platforms

The majority of CTI APIs to date have been limited by their design to mainframe computers and minicomputers, and were difficult to adapt to use in personal computer platforms. Some early attempts to integrate telephony into personal computers required specially-designed PC hardware; an example is the "TeleCompaq" computer, that included an integrated handset and other special telephony hardware. In many cases, this specially-designed PC hardware was not expandable, insufficiently powerful for general computing needs, and introduced incompatibilities with some operating systems software and applications. This experience has shown that to be successful, integration must be based on standard PC hardware, and must use standard software and software development tools. TAPI builds on the Microsoft Windows foundation, using standard PC hardware and software, preserving users' investments, and allowing the flexibility to which PC users have become accustomed.

TAPI 2.0 is Flexible and Scaleable

The Microsoft Windows foundation and TAPI 2.0 combine to provide the flexibility and scalability upon which a rich and diverse CTI industry can thrive. With TAPI, developers can create applications ranging from simple desktop telephony helpers to complex call-center packages that support thousands. In addition to supporting the widest array of call models, TAPI allows users to run several telephony applications simultaneously on a client or server PC. Of course, at the heart of TAPI 2.0 is the abstraction of the hardware layer.

Abstracting Hardware for Flexibility and Network Independence

One of the major problems in connecting the computer to the telephone is the enormous complexity and diversity found in the underlying telephone network. Virtually all of the telephony APIs in the industry today have been promulgated by an individual switch manufacturer, and require use not only of their PBX but also of interface hardware or server software sold by the same manufacturer. Other APIs, while available on a variety of hardware platforms, have nevertheless been generally restricted to a single networking environment, such as PBXs or ISDN, and are not portable to other environments such as "Plain Old Telephone Service" (POTS) or Centrex. This has meant that the market for any particular application written in conformance to these APIs has been small, since different versions (often with quite

different user interfaces) were required for each switch or network.

The Windows Telephony API abstracts the vast array of proprietary hardware and provides standard programming interfaces that allow users and application developers to take advantage of the many capabilities and services of different telephone systems. TAPI is designed to permit the implementation of drivers that work with any network topology and call model, isolating the application from the complexities of any particular network (for example, Q.931 signaling in ISDN).

This means application developers are isolated from the complexity and variability of the underlying telephone network and no longer have to "hard-code" their applications to a particular system's signaling or message set requirements. Telephone network providers can tap into a broad range of applications, thereby driving network utilization. Customers can develop or purchase applications and protect their investment as applications are used with different phone systems, in different offices, or even at employee homes.

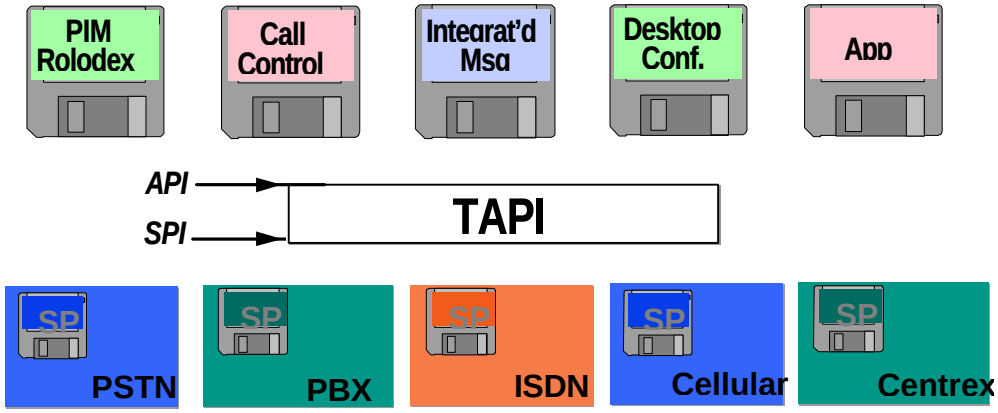


Figure 1. The Telephony API approach—abstract the hardware from the software.

Applications developers write to the Applications Programming Interface (API), while network providers write to the Service Provider Interface (SPI). This separation, similar to the way printer or display drivers work under Windows, allows developers to focus on their application and not the plumbing required to connect to a specific network. Developers can focus on their primary task, namely developing the application itself. As a result of network independence, TAPI supports PBXs, key telephone systems, ISDN, the analog PSTN, cellular, Centrex, and other types of telephone networks.

Providing Connection Model Independence There are four basic ways to integrate a PC and a phone. TAPI supports all of the models, ensuring maximum flexibility for how applications are deployed. Again, customers may have different configurations for different employees in different offices—even for use in homes. The same application can be used in different configurations, because the application vendor doesn't have to assume a particular configuration is being used. TAPI

hides the actual connection from the application (similar to the fact that whether a printer is local or network connected is irrelevant to a Windows application that prints). This allows TAPI to enable a universal telephony client. *Figure 2* shows various ways in which a computer can be connected to the telephone network—and TAPI supports them all.

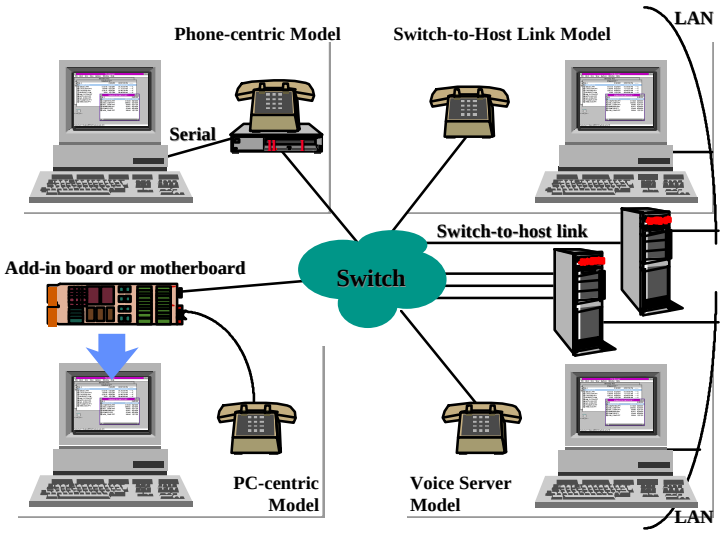


Figure 2. Connection model independence.

Phone-Centric This model involves an external adapter or telephone that is connected through the PC's serial port. It is easy to configure, but may be limited in how much information and control it supports from the PC. The new generation of modems support voice in addition to data and fax (the AT+V or IS-101 standard), which will allow external modems to serve as a telephone designed as a PC peripheral. Voice modems are the cheapest, highest volume hardware to bridge the PC and the phone. While this configuration works well for a single user or a small group, it is not usually the most economical configuration path for a larger group or entire enterprise because of the expense of special CTI-enabling hardware at every desktop that is dedicated to each PC-to-telephone combination.

PC-Centric In this model the phone line terminates on a card in the PC or on the motherboard itself. Many new PCs come with a phone jack built-in. This model maximizes the information and control delivered to the PC. As signal processing capabilities become more and more common, this functionality will be increasingly standard on every PC. One example of this configuration is the evolution of the multimedia audio card into a multimedia communications card with support for standards like UniModem V. In addition to the audio capabilities delivered today, these cards also support voice, data, and fax communications for about the same price as today's audio board. This configuration, like the previous setup, works well for one person or a small group, but does not necessarily scale well to large groups or across an enterprise due to the hardware at each desktop.

Switch-to-Host Whereas the first two models involve a physical connection between the PC and phone at the desktop, in this model there is only a *logical* connection between them, established via the LAN. There is a physical connection, but in this case, it is between the phone system and a server operating on the LAN. This model is applicable to PBXs and key systems, though it has limited application for public networks like analog, ISDN, Centrex, and cellular, as well. The "switch-to-host" link is an intelligent link which provides real-time status on calls handled by the PBX or key system to a server connected by this special link. The server can process this information and control the PBX based on events it monitors, such as an incoming call. The server can control the phone system on behalf of individual PCs.

This model allows calls to be controlled, but has no provision for moving information from the PC out over the phone network, such as a fax or desktop conferencing session. This model is primarily suited to today's call center market, where being able to transfer a call to a supervisor is sufficient. This configuration also allows, for example, a customer database record to be instantly placed on the PC screen at the same time a call is routed to the agent's phone as a means to reduce the amount of time to handle a reservation or other customer service action. This enables a business to handle more calls with fewer agents at a lower cost while delivering better customer service, so the business impact is substantial.

Because of the historically-high costs associated with these solutions, they have typically been found rarely outside very large corporations or mission-critical call centers where the investment and customization was warranted. Now, with an increasing number of off-the-shelf CTI solutions, this type of configuration becomes much more affordable allowing an organization to economically deliver the time savings and other benefits of CTI to a broader population of users within an enterprise. The primary challenge most IS and telecom managers experience in this configuration is due to the complexity associated with setting up on the PBX or key system some of the parameters to make the intelligent link work properly. The hardware abstraction of TAPI 2.0 makes it easy to do incremental upgrades. Businesses can add value to their current legacy phone system by adding CTI enablers and applications.

Voice Server This model is also a logical connection, but instead of, or in addition to, a special switch-to-host link, ordinary phone lines are used between the server and the PBX or key system. With this setup, information can be routed to the server via the LAN and can then be transmitted over the phone network. Fax servers and voice mail systems use this configuration, and it will become increasingly common with the move away from the "mainframe"-like switch-to-host model because the voice server is more flexible and supports a greater variety of applications. The switch-to-host link model, covered in the previous paragraphs, is limited to simple signaling, so that when an application requiring access to the media stream is desired, one of the these other connection models has to be layered on top, resulting in a redundant infrastructure. The voice server model is the "PC" approach to the switch-to-host link's "mainframe" model.

The voice server model offers some rich possibilities to extend CTI support throughout an enterprise economically and with less complexity. In contrast to the switch-to-host set-up covered previously, in which a significant portion of effort and

expense can be invested in making the intelligent link actually work; in the voice server model, those dollars can be applied to a more integrated server-based solution using well understood and less expensive voice trunks or lines to connect the phone system to the server. In some cases, this configuration—including the server hardware—can be less expensive for a total solution than the switch-to-host link setup, yet yield far more applications flexibility and tighter, inherent integration between the voice applications and other applications.

The Internet TAPI's extensibility includes the Internet, as the figure below indicates. One of the more recent changes impacting CTI is the role the Internet can play in adding value to CTI applications. TAPI's architecture does not relegate it to supporting only public switched telephone network (PSTN) based solutions. Some of the more innovative uses of the Internet with telephony involve the weaving together of the Internet, the PSTN, and legacy phone systems for a range of applications, such as, sending fax or voice messages across the Internet or enabling a call management application to route a call over either the PSTN or via the Internet based upon user or administrative preferences.

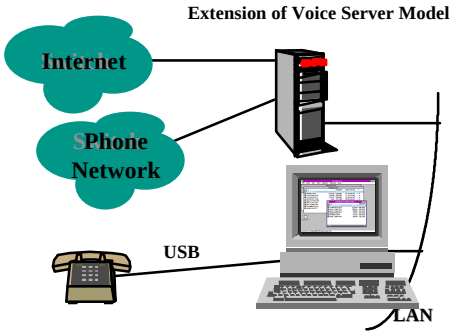


Figure 3. TAPI can be used to enable applications which work with Internet or PSTN.

TAPI, because it is network independent, allows the Internet to be used for telephony functions. Microsoft sees this as becoming increasingly relevant in the future, as companies try to make more use of the Internet for virtual private networking. Quality of service improvements that are coming, with QoS-based routers, operating systems, and such, will enhance the ability to do real-time telephony across the Internet even prior to the advent of full ATM switching. Because ATM is inherently QoS-based, it will provide even greater support for TAPI-enabled real-time applications. So the Internet will increasingly become a carrier service for companies to take advantage of for virtual private networking—for both data and voice.

Universal Serial Bus The Universal Serial Bus (USB) is a new 12-megabit per second capacity bus structure which will soon be built into every Intel motherboard shipping in the near future. It replaces the standard serial port, providing plug-and-play and daisychain functionality that will free users from having to open their computers to install peripheral cards, set jumpers, and change dip switch settings. The USB, an open industry standard, was developed in part to accommodate the host of new peripheral products coming to market—including those for telephony. In all, USB can connect up to 127 different devices to a single PC.

The USB provides a very good way of interconnecting the phone and PC, particularly at the desktop, and because it supports up to 12 megabits per second of instantaneous bandwidth, it will be more than sufficient for voice telephony.

Supporting Multiple CTI Applications Simultaneously There is a variety of applications that can vie for use of the phone system. TAPI provides a mechanism for these applications to coexist and share a single phone line or multiple phone lines. Thus, one application could be waiting for incoming faxes while another wants to dial out to an on-line service. Historically, the presence of the first application would prevent the second from using the phone line and result in the dreaded "Sorry, COMM Port X is in use by another application" message. Yet the device connected to the phone line is actually only in use when another application is on a call. Because TAPI resides at the operating system level, as long as the fax application is not actually in the middle of receiving a call, the on-line session can be established.

Making telephony support an operating system-level service becomes even more important when the *incoming* call problem is considered. When the phone rings, there is no way to determine if it is a voice call, data call or fax call. TAPI provides a mechanism for calls to be handed off to the right application based on the type of call once that call type has been quickly determined upon answer.

Building Upon a Solid Multipurpose Operating System TAPI-enabled applications benefit from the solid multipurpose Windows platform, including the rich set of Windows APIs and the power and flexibility of ActiveX Controls.
Windows NT 4.0 is a state-of-the-art operating system with 32-bit preemptive multitasking, multithreading, excellent security, support for a variety of processors and system configurations, integrated administration, and other complementary APIs and tools.

Windows offers a rich set of services desired by developers, and there is a broad array of available development tools which can be used to develop and test Windows NT applications. Windows is a portable operating system, which ensures applications can be easily migrated to whatever CPU architecture offers the best price performance. Similarly, with built-in support for symmetrical multiprocessing, applications can start on a single CPU machine and grow as processing requirements grow without modification to the application.

Serving as Telephony Client or Server Because TAPI 2.0 is an integrated service of the Windows NT 4.0 operating system, Windows NT 4.0 can function as either a telephony client or a telephony server. In fact, Windows 95 and Windows NT Workstation 4.0 are now the only *client* operating systems and Windows NT Server 4.0 is the only *network* operating system with built-in telephony support. There is no extra charge for this telephony support with Windows.

Providing a Secure Platform Because an application can only be as secure as the operating system upon which it runs, Windows NT was designed from its very core to be a secure system. The National Computer Security Center has determined Windows NT security to have satisfied all specified requirements for a rating of C2. Windows NT provides a secure platform for protecting telephony applications, yet the security features are easy to use and unobtrusive, so they don't impede users.

Building Upon other Applications Windows NT offers a rich set of "applications building blocks" developers can exploit to provide value-added solutions. This is one

of the factors creating the strong momentum Windows NT has earned in the marketplace. These are high-volume, world-class, affordable, off-the-shelf software and hardware products that are readily available for use with Windows NT and Windows 95. For example, Microsoft SQL Server™, a powerful client/server database application, Microsoft Exchange, a messaging and scheduling application, and Microsoft SNA Server, a leading mainframe-to-server integration application, *all* provide a broader platform upon which third-party developers can add value and integration with computer telephony. Other application building blocks are also available from other companies.

Similarly, off-the-shelf, PC-based voice processing components for Windows NT are available from a variety of vendors. The result is that developers and customers can exploit the favorable cost curves and competition found in the PC industry. This speeds time to market and offers more efficiency and innovation than is found in a market built on proprietary systems.

TAPI 2.0 is Comprehensive

TAPI 2.0 and the Windows NT and Windows 95 operating systems provide a comprehensive foundation for telephony applications, which allows developers to create a wide range of products. The exhaustive list of telephony features supported by TAPI means that a developer can telephone-enable virtually any general purpose application. And TAPI provides unrivaled supplementary and extended support for . telephony-centric applications.

TAPI 2.0's support of Unicode makes it easier to create applications that work globally. And, of course, adding to the comprehensiveness of the Windows telephony platform are the APIs and other complementing support from WOSA elements. Additionally, ActiveX Controls for telephony, provided by a variety of vendors, allow corporate developers to put together powerful telephony applications even faster and more easily using tools they already understand.

Providing Four Levels of Telephony Service

To provide developers with maximum flexibility, telephony support can be implemented at four levels with TAPI:

- Assisted Telephony
- Basic Telephony Services
- Supplemental Telephony Services
- Extended Telephony Services

Assisted Telephony Assisted Telephony is a small set of functions that provide very basic telephony functions to primarily non-telephonic applications. This valuable feature is designed to make the establishment of voice calls and of media calls available to any Win32-based application, not just those dedicated to telephonic functionality. In other words, Assisted Telephony lets applications make telephone calls without needing to be aware of the details of the services of the full Telephony API.

Assisted Telephony extends telephony to word processors, spreadsheets, databases, personal information managers, and other non-Telephony applications. For example, adding an Assisted Telephony function (tapiRequestMakeCall) to a spreadsheet lets users automatically dial telephone numbers stored in the spreadsheet (or in a connected database). Assisted Telephony is the easiest and most

efficient way to give telephonic functionality to non-telephonic applications.

Basic Telephony Services Basic Telephony Services, rather than Assisted Telephony, is used for applications whose telephony functionality goes beyond just the making or receiving of a call. The complete TAPI defines three levels of service, of which Basic Telephony is the first.

Basic Telephony Services are a minimal subset of the Win32 Telephony specification. Since all service providers must support the functions of Basic Services, applications that use only these functions will work with any TAPI service provider. The functionality contained in Basic Telephony roughly corresponds to the features of "Plain Old Telephone Service" (POTS) such as make call, hang up, and answer call.

Basic Services include:

- Address translation
- Making calls
- Call states and events
- Answering calls
- Dropping calls
- Call handle manipulation
- Assisted telephony server

Today, many programmers will use only the services provided by Basic Telephony. And, at a minimum, service providers are be designed to support all of the Basic Telephony functions. Other application developers, such as those writing code for PBX phone systems, will need the functions of Supplementary Telephony.

Supplementary Telephony Services Supplementary Telephony Services include the bulk of what TAPI offers—a very complete set of functions to enable a wide range of powerful, easy-to-use telephony applications. In Win32 Telephony, supplementary functions are those whose form and functionality have been defined by the API description, but which are not required in Basic Telephony. They are functions that developers of telephony applications and service providers may choose to implement to suit the design of their custom products. So, in contrast to Basic Telephony functions, Supplementary Telephony functions are optional.

Supplementary Services include:

- Hold
- Transfer
- Conference
- Forward
- Park
- Pickup
- Completion
- Accept, redirect, reject
- Secure from interruptions
- Generating digits and tones
- Digit and tone monitoring
- Media mode monitoring
- Media stream routing and control
- User-user info
- Change call parameters

- Phone terminal control

Extended Telephony Services TAPI contains a well-defined API extension mechanism that allows service-provider vendors to extend the Telephony API using device-specific extensions. Extended Services (or Device-Specific Services) include all extensions to the API defined by a particular service provider. Since the API defines the extension mechanism only, the definition of the Extended-Service behavior are completely specified by the service provider.

TAPI's extension mechanism allows service-provider vendors to create functions not directly defined by the Telephony API. This flexibility provides the power developers need to create their own custom, value-added marketplace solutions.

A Comprehensive Level of Control TAPI 2.0's comprehensiveness is carefully thought out. For example, it provides developers with an excellent level of control—while guarding against allowing excessive controls that might limit applications to specific hardware settings.

Some of the more difficult decisions to be made by the architects of a telephony API relate to the level of control over specific hardware that is to be permitted to applications.

For example, developers of applications that use telephony boards designed to the MVIP specification are accustomed to controlling which "timeslot" on a high-bandwidth channel is carrying the signal for a particular call to and from the board in question. Such applications would not be easily portable to, say, a modem or PBX phone. An API that required (or even encouraged, by making available) such detailed control would not facilitate the development of highly portable shrink-wrapped commercial telephony applications. The precise control of such low-level details is more appropriately assigned to an interface between a low-level device driver and the hardware, rather than the relatively high-level interface between the operating system and application programs.

On the other hand, if the API gives applications insufficient control of call details, some beneficial functionality may be lost. Users complain of being unable to perform certain functions through their software that are easily performed through the telephone itself. An example of this is in the area of establishing and controlling multiparty audio conference calls. It is possible to create a very simple conference call API, but this may not be flexible across a variety of platforms (PBX and Centrex conferencing models differ in significant ways), to allow the addition and removal of arbitrary calls from the conference, and so forth. The ability to control certain low-level elements such as the timing and duration of Touch-Tone (DTMF) signals is also important to compatibility with other equipment.

Using Media-awareness A successful CTI API must implement basic call control features—but all this brings to users is, fundamentally, the ability to move the telephone user interface from the keypad to the PC screen. There are certainly benefits in this alone, such as allowing the interface to be context-specific (showing only the functions actually available at the moment) and eliminating the need to remember arcane multidigit feature activation codes and key-press sequences. Most CTI APIs to date have been limited to call control. Realization of the most significant benefits of computer telephony integration requires, however, the inclusion in the API of access to the "media stream"—the information content of the call.

Media-awareness comes in several flavors. First are those features that are unique to the telephony environment, such as the monitoring and generation of dual-tone multifrequency ("Touch Tone") digits and call progress signals. Since these features are not included in any other general-purpose operating system API, it is appropriate for the telephony API to include them directly.

Second are features that have been part of PC operating systems for many years and for which there are well-established APIs. These include the playing and recording of sound and the sending and receiving of data of various kinds. Replicating these features in the telephony API would be redundant and confusing to programmers, and would inhibit the use of the infrastructure and tools that already exist for use with those other APIs. For example, developers of modem communications software are much more likely to migrate their code from directly controlling the modem to using a telephony API for call control if their investment in finely-tuned terminal emulation and file transfer code, which uses existing operating system APIs for data transfer, is preserved. Developers of telephony applications appreciate being able to use higher-level audio-related features such as speech recognition, text-to-speech, and mixer control, which have already been developed for use with local audio equipment (microphones and speakers), to manage telephone calls as well. A well-architected CTI API must allow telephony device drivers to be associated with linked device drivers that are accessed via these media stream APIs, and provide means for applications to become aware of the device identifiers associated with the device to be used via these other APIs to give access to the media stream of the call.

Managing Media Streams Windows TAPI supports telephony as a service at the operating system level, giving it the ability to handle media streams and support multiple applications.

There are two types of information transmitted over the phone network. There is information that is carried from one endpoint to another, such as a voice conversation, a data modem session or a fax. This is known as *a media stream*. The other type of information is *signaling*. Unlike a media stream, signaling only moves between an endpoint on the network and the network itself. Signaling comprises the housekeeping instructions to and from the network, such as dialing or transferring a call or delivering Caller ID.

TAPI provides access to the signaling for setting up calls and managing them, as well as preserving existing media stream functionality to manipulate the information carried over the connection TAPI establishes. This allows applications to not only dial and transfer calls, but also to support fax, desktop conferencing or applications that use the telephone set dial pad to access voice-prompted menus.

TAPI's tight integration with other Windows APIs allows it to seamlessly invoke other APIs and Win32 functions, such as the Network Driver Interface Specification (NDIS), COMM, Wave Audio, MIDI, and MCI to handle media, sound, and video.

Evolving the Telephony API Microsoft has steadily evolved its Telephony API, as it has evolved its Windows operating systems. TAPI's evolution has consistently been guided by the active involvement of leaders from throughout the telecommunications industry, including close consultation with independent hardware and software vendors, and service providers. The result has been an ever more

powerful and flexible telephony platform for developers and end users.

TAPI 1.3 TAPI 1.3 was released as a standalone SDK in November 1993. It focused on call-control applications such as PBX support, but handled only 16-bit applications and didn't provide "dialing properties."

TAPI 1.4 TAPI 1.4, released as part of Windows 95, provided substantial enhancements, including:

- Support for 32-bit applications
- Plug-and-Play convenience
- Common dialog for dialing properties
- Access to country list
- Get/Set application priorities
- Provider-initiated conferences
- Other capability/status extensions
- Universal modem driver

Both TAPI 1.3 and 1.4 provided excellent first-party call control. Some companies created client/server applications, but had to provide additional "plumbing" on their own—something that is being solved with TAPI 2.0.

TAPI 2.0 TAPI 2.0 is built into Windows NT Server 4.0 and Windows NT Workstation 4.0 (and will soon be made available for Windows 95). For developers, the TAPI SDK is integrated with the Win32 SDK developer's platform. That platform is part of Microsoft Developer Network (MSDN), for Windows 95, Windows NT Workstation, and Windows NT Server. The SDK includes headers, libraries, sample code, and documentation. TAPI 2.0 includes these enhancements:

- **Full 32-bit architecture.** All core TAPI components are now based on Win32, with full support for non-Intel processors running Windows NT Server 4.0 or Windows NT Workstation 4.0. There is also full support for symmetrical multiprocessing, multithreaded applications, and preemptive multitasking.
- **32-bit application portability.** Existing applications that are based on Win32 full TAPI or TAPI assisted telephony support and currently run on Windows 95 using TAPI 1.4, will run on Windows NT Workstation 4.0 or Windows NT Server 4.0 on the Intel x86 family of microprocessors without manipulation or recompilation.
- **16-bit application portability.** Existing applications that are based on Win16 full TAPI or TAPI assisted telephony support and currently run on Windows 95 and Windows 3.1 using TAPI 1.3, will run on Windows NT Workstation 4.0 or Windows NT Server 4.0 on the Intel x86 family of microprocessors without manipulation or recompilation.
- **Unicode support.** Win32 applications can now choose to call the existing ANSI TAPI functions or to call the new Unicode versions of functions. Unicode support makes it easier and faster for developers to create products and localize them to different languages. Unicode is a 16-bit, fixed-width character encoding standard that encompasses virtually all of the characters commonly used on computers today—this includes most written languages, plus publishing characters, mathematical and technical symbols, and punctuation marks.
- **Expanded feature support for call center applications.** TAPI now supports an expanded set of features to enable more complete call center operation with Windows. Here is brief list of some of the new call center features supported with

TAPI 2.0:

- ACD queues
- Agent log-on, state and activity monitoring and control
- Predictive dialing
- Call routing
- Call data
- Call treatment
- Message waiting and display control without using telephone set functions

- **Registry support.** All telephony parameters are now stored in the Windows registry. Telephony service providers and all stored parameters can be updated across the LAN, making it easier to set up, use, and manage computer telephony solutions with Windows.
- **Quality of Service (QoS) support**. Applications can request, negotiate, and renegotiate quality of service (QoS) performance parameters with the network and receive an indication of QoS on inbound calls and when QoS is changed by the network. The QoS structures are binary-compatible with those used in the Windows Sockets 2.0 (WINSOCK) specification. Improved QoS support reduces or eliminates latency and other negative characteristics for applications, especially voice and data applications, over various networks. Going forward, QoS support enables the easier migration of services and applications previously acceptable only over the public switched telephone network to be supported over a LAN or WAN based on TCP/IP, ATM, or other technology.
- **Enhanced device sharing**. Applications can restrict the handling of inbound calls on a device to a single address, to support features, such as distinctive ringing, when used to indicate the expected media mode of the inbound calls. Applications making outbound calls can set the device configuration when making a call. This makes computer telephony support in the home or small office even more useful.
- **NDIS TAPI capability**. The support previously provided in Windows NT 3.51 for ISDN Wide Area Network (WAN) miniports under Remote Access Service (RAS) is preserved. NDIS WAN miniport drivers are supported under a kernel mode service provider without modification. TAPI is used to support essentially all dialing functions within the Windows NT operating system, which includes control of WAN cards for remote access. Support for NDISTAPI drivers means it will be easier to find WAN cards that work with your Windows-based PC.
- **Several changes and additions to TAPI functions**. Many new TAPI functions and messages are now available with TAPI 2.0. In addition, several functions and messages already supported by TAPI 1.4 were changed in some measure to make them easier to use and more consistent in their operation. These enhancements make TAPI, already an excellent platform for telephony applications, even more comprehensive. For a complete list of these new functions and instructions regarding their programming use, consult the Microsoft Developer Network resources.

Building in Extensibility Part of evolving an API is to build into it a solid path for extensibility. Not even the most rich and robust of CTI APIs will ever encompass every feature that might be created by the telephony industry. So a CTI API must grow over time to include new features that emerge and gain acceptance. TAPI

was designed from the start to grow—to allow new fields to be added to data structures in a straightforward and backwardly-compatible manner, for new messages and events to be created, and so on. TAPI also allows a mixture of old and new drivers and applications to coexist through a version negotiation mechanism.

TAPI also allows developers of telephony drivers to include unique or advanced features in such a way that the presence of these features can be detected by applications that have been designed to use them, without blocking older applications that aren't able to take advantage of them.

Looking Ahead In the months following TAPI 2.0's release, Microsoft plans to release additional enhancements, which will include:

- Windows Telephony Service (TAPISRV) extensions for client access.
- TAPI 2.0 for Windows 95 clients, with backwards compatibility for 16-bit TAPI Service Providers (TSP). This will allow Windows 95 clients to take advantage of the built-in call center application hooks.
- Client service provider for Windows 95 and Windows NT 4.0 to make client/server telephony applications even easier to build.
- Remote service provider, to enable even faster development of client/server telephony applications.
- Remote administration tools, to make it easier for companies to set up and maintain client/server applications. Tools will include user access control, workstation/device associations, service provider configuration, status, events, and other reports.

Continuing to evolve the Telephony API is part of Microsoft's commitment to always providing the best platform for whatever services and applications the market creates a need for. Microsoft works constantly with customers and developers, seeking insights into how the platform should evolve.

TAPI 2.0 is Open and Written to Industry Standards

TAPI was defined with substantial, ongoing input from the telephony and computing industries from all over the world. These efforts, including extensive input from independent software and hardware vendors, have made TAPI a truly open industry standard.

TAPI works with a variety of PC processors, runs on industry standard hardware platforms, and as has been noted earlier, the hardware abstraction of TAPI is a huge contribution to openness. Its switch-fabric independence means TAPI-compatible applications can be run on a wide variety of PC and telephony hardware and support a variety of network services.

TAPI developers and their applications also benefit from the openness and industry standards that are such a foundation of the Windows NT and Windows 95 operating systems and complementing WOSA elements.

Programmability Because TAPI is based on the Windows operating system, telephony applications can be developed and tested using standard tools that are widely available. TAPI developers can program in C, C++, Visual Basic®, or any of a number of other popular programming languages. This frees developers from having to learn the Hayes "AT" command set, which historically has been the dominant telephony programming model.

The AT command set is useful, but suffers from limitations including bandwidth, a strict command-response model, in-band notifications and commands (call control commands and messages cannot be easily invoked during a call), a limitation to the 7-bit ASCII character set, and a lack of standardization across the industry beyond the most basic functions.

Microsoft's Windows 95 includes the "Universal Modem Driver," a Windows Telephony service provider intended to be able to control virtually any AT command set-based modem. Its modem configuration information consists of 50 files totaling over 1.3 megabytes of data and defining interoperability with over 800 modem models in the USA alone—and this just covers the most important features of the most popular modems. The variations between products claiming compliance to such a long-standing industry standard is striking.

Probably the greatest limitation with the AT command set is that it uses a completely different syntax and structure from the programming models used by the majority of application developers today for most of their work, such as C and C++. The need to learn an entirely new programming environment limits the utility of the AT command set. Likewise, any telephony API that departs in significant ways from the most familiar programming environment will have a longer and higher learning curve, be more prone to errors and confusion, require additional training and support, and have lesser acceptance among application developers.

With TAPI, developers can easily take advantage of other elements of the operating system such as related data transfer APIs for access to the media stream of the call. Such integration with the operating system allows for the creation of higher-level constructs such as Visual Basic custom controls, OLE controls, and C++ object libraries that are closely tied to other elements in the system.

Usability In addition to the solid usability and intuitive interfaces that are part of the Windows environment, TAPI allows developers to create usability features specific to telephony such as dialing location independence. TAPI-enabled applications can include features to help users create and manipulate sets of dialing properties for the locations from which they use their computers.

In Windows Telephony, users can store telephone numbers in their applications and address books in a "location independent" form, such as the notation specified in ITU-T Recommendation E.123 (which calls for a string in the form "+*cc (ac) ln*" where "cc" is a country code, "ac" is an area (or city) code, and "ln" is the local subscriber number within that city). The user configures the telephony service for the dialing requirements of their location, including the country code and area code of the location, area codes and exchanges that are local or toll calls, outside line access codes, call waiting prefix codes, tone or pulse dialing, telephone credit card access numbers, and so forth. Applications can then request TAPI to "translate," or convert, the number in international (location-independent) form into the series of dialable digits and other dialing instructions needed to complete the call at the current location.

TAPI can store many such sets of location dialing properties for users who frequently move their computers between office, home, and other remote locations. The location independence enabled by this CTI API service facilitates other important enterprise services, such as "global address books." For example, a corporation

may make available (through a Messaging API address book service provider) a single enterprise-wide directory of phone numbers. In a multinational corporation, if the telephony API provided no number conversion, the numbers in this global directory would have to be manually dialed by each caller, extracting the portion of the number relevant to the location and adding such prefix and suffix digits as are needed to complete each call. With location-specific dialing properties, however, telephony applications software can easily apply local dialing procedures and thereby automatically dial calls from the global address book.

Another example is that of a multinational service, such as CompuServe or MSN™, The Microsoft Network that publishes a list of its access numbers in hundreds of cities worldwide. Without the location independence offered by phone number translation based in the CTI API, users would have to manually add prefixes and other dialing instructions to the access numbers published by the network; location independence allows for the distribution of a single global access number directory from which the correct access number can be properly and automatically dialed, so long as the user has correctly specified their local dialing instructions. Applications can also take advantage of knowledge of the selected location to automatically use the correct access number.

TAPI will also use operating system services such as Plug and Play to automatically detect the attachment of new devices and make them available to applications. Inserting a PCMCIA modem, or powering up the system with a new telephony device attached, should result in the system automatically detecting the new hardware, loading the correct driver into the telephony environment (or requesting the user to insert a diskette containing the driver). Telephony drivers that consist of software only, such as the client portion of a client/server LAN-based system, should be able to be installed, configured, and upgraded remotely over the LAN using software such as Microsoft Systems Management Server, and be made immediately available for use.

Using ActiveX Controls ActiveX is a set of technologies that enables software components to interoperate in a networked environment, providing end users with a richer, more interactive experience. The ActiveX technologies are enhancements to OLE, Microsoft's component software technology, which has become a well-established, industry standard since the introduction of OLE in 1994. ActiveX Controls are small, efficient modules that implement specific, specialized functions.

Software companies around the world create ActiveX Controls, including companies such as Borland, ORACLE and Sybase/Powersoft. The result has been the creation of more than 1,000 ActiveX controls, which are available for developers to use as building blocks in creating applications.

For telephony developers, ActiveX Controls from companies such as Stylus Innovation, Technically Speaking, and Pronexus provide plug-in components that can make the job of creating applications much easier. This allows a broader population of developers to concentrate on the unique features of their programs, rather than the plumbing. ActiveX also allows developers to work with a rich variety of programming languages, including Visual Basic, Visual C++®, Borland Delphi, Borland C++, Java, and Java-enabled tools. These graphical user-based programming tools speed up the development process substantially and provide programming and customiza-

tion of telephony that have long been desired but which were not possible before now.

PROVIDING THE FOUNDATION FOR RICH APPLICATIONS AND SOLUTIONS

The robust and reliable Windows platforms—whether Windows 95 or Windows NT 4.0—combined with TAPI, and other key APIs and Win32 functions, such as SAPI, MAPI and Wave Audio, along with the powerful addition of ActiveX Controls, provide the ideal platform for rich telephony applications.

Microsoft's goal is to make the telephone as common a PC peripheral as the monitor or printer. TAPI allows applications developers to take advantage of the telephone network to create a world of telecommunications applications.

This combination of Windows with TAPI and related Windows APIs enables a spectrum of client-only, client/server, and server-based telephony uses and applications, including:

- Visual call control
- Exploiting voice as a data type
- Telephone-enabled productivity applications
- Unified messaging
- Auto attendant
- Desktop conferencing
- Vertical applications
- Call center applications with IVR and predictive dialing

Visual Call Control

Visual Call Control brings the point and click ease of computers to the world of telephones. Instead of requiring a user to remember cryptic codes or figure out how to use a phone covered with a multitude of buttons, phone operations can be performed from the rich user interface of the PC and integrated with PC-based information such as personal directories or databases. Simple operations like programming speed dials and transferring a call become easy (no more "Gee, if I lose you..."). Other network features like Caller ID can also be exploited because TAPI provides a standard delivery mechanism for applications developers.

Exploiting Voice as a Data Type

The PC can serve as a smart answering machine, not just taking messages, but screening and forwarding calls based on user preferences. Further, interactive response systems, like the telephone-dialpad-based menu systems used by many banks, can be easily constructed on the PC, making this capability easily available to departments, small businesses, and even individuals. Finally, speech recognition will allow users to call their computer from the car or airport and have the PC retrieve information or perform tasks for them—even read incoming e-mail aloud over the phone line.

Telephone-Enabled Productivity Applications

Existing productivity applications like personal information managers or databases can benefit from integration with the telephone network. At the simplest level, any phone number stored on the PC can be dialed with a double-click or speech command.

Unified Messaging

Rather than trying to manage electronic mail from the PC, voice mail from the phone, and faxes from another source, a single graphical interface can be used to manage, prioritize, retrieve, and archive all types of messages. Voice messages can be retrieved in any order with the click of a mouse, instead of having to wade through every message in sequential order. Faxes can be easily sent and received from the PC, without ever turning into paper.

Auto Attendant

"If you know the extension of your party, enter it now. If you would like to speak to an operator, press two." Auto attendants are being used more and more by businesses large and small to handle initial routing of incoming telephone calls. TAPI makes it easy to create and use auto attendant applications, meaning that small businesses, and even individuals, can take advantage of it. This is one way that a small business can appear to the outside world to be a much larger business. By making use of these kinds of technologies, which people tend to associate with large businesses, they will enhance their customers' perceptions of them. TAPI will provide those functions at a level which is very affordable for even a small company. Auto attendant also saves labor on the part of receptionists and others, which is another cost savings. It is a way of automating certain routine tasks, such as rudimentary queries. These can often be predicted, and an auto attendant can often handle them. Human labor can then be allocated to other customer service tasks.

Desktop Conferencing

The ubiquitous phone call is evolving from just voice to data conferencing which mixes voice and data, and ultimately to video conferencing. Data conferencing works on today's telephone networks and allows people to share applications or use a common whiteboard while they talk in order to collaborate and solve problems together. As network bandwidth expands, video conferencing will become more widespread. Further, these applications will increasingly work well independent of the network infrastructure used—for example, a public switched telephone network, the Internet, or some combination of the two. Microsoft's NetMeeting is an excellent example of this type of collaborative communications tool which spans the Internet and the PSTN.

Vertical Applications

Historically, everyone in an organization, or at least everyone within each defined user group within an organization, has had essentially the same set of phone features. Now, with the capability to meld the phone system with the information system through an open and easily programmable PC platform, there will be an emergence of new integrated communications systems, tailored for call centers, law offices, retailers, hospitals or any other market segment with unique communications requirements.

Call Center Applications

The telephony server is notified of an incoming call in an inbound telemarketing or

customer service environment. The application on the server can examine the Caller ID associated with the call (all 800 calls, for example, provide Caller ID-type information) or use account number information entered by the caller and use that information to do a look-up in the customer database. Upon return of the customer's information, this data can be routed over the LAN to the next available customer service agent, or even a particular agent who regularly works with that customer. The application on the server can then direct the call to the appropriate agent who has already received the necessary information to deal with the call.

Historically, call-center applications like this have been built upon relatively expensive, proprietary systems that are challenging to customize and integrate with existing information systems. By making these applications entirely software-driven, they can be developed more quickly, more cheaply, and be more tightly integrated with PC-based information systems. This can result in ever more sophisticated systems for customer management or making call-center capabilities accessible to smaller organizations.

Help Desks

Help desks are a special case of the call center that serve users within an organization. The same applications used to serve external customers can be applied to serving internal customers.

Predictive Dialing

In an outbound telemarketing environment, a telephony server can rapidly dial a list of desired numbers and when a connection is detected with a live person, immediately route the call to a customer service agent. Because only a fraction of calls result in connections with people, as opposed to those that are busy, have no answer or are picked up by an answering machine, this application can dramatically improve the efficiency of outbound telemarketing by limiting time spent on non-productive calls.

Interactive Voice Response

These systems allow users to create voice menus that callers can traverse with the telephone dial pad, using them to access information or execute transactions. Like many other telecommunications systems, Interactive Voice Response (IVR) systems are moving away from proprietary systems to PC-based telephony servers.

Voice Mail

Voice mail is also moving to PC platforms to exploit less expensive disk storage and integrate better with e-mail systems to offer a single, "Universal Inbox" containing e-mail, voice mail, and fax messages.

Routing

There are a variety of call routing services that can be implemented that redirect calls according to user or system preferences.

PC PBX

TAPI can be used to bring the entire telephone switch inside the PC and provide a well-integrated, single box solution.

And Beyond

Even more exciting are the products that haven't been conceived of yet. But as soon as the telephone network is accessible to an applications developer with a great idea, there are certain to be applications that transform the way we communicate.

All of these applications will let individuals communicate more effectively and efficiently. The PC as a communications platform gives users more control over who they talk to, when they talk to them, and what they talk about. Organizations can empower their members with better communications tools.

BENEFITING FROM THE WINDOWS TELEPHONY PLATFORM

The big benefits for developers, vendors, and end users will be creation of the kind of personal telephony applications that TAPI and Microsoft Windows enable. Next-generation telephony will help end users get the most from telecommunications systems, allowing them to more efficiently manage their voice calls and control their data-transfer operations. TAPI can bring this efficiency to any application—database manager, spreadsheet, word processor, personal information manager—any application that can benefit by sending and receiving data through the telephone network. This will bring immense benefits to all segments of the CT industry, ranging from software developers and hardware manufacturers, to resellers, telecom managers, and business users.

Benefiting Application Developers

Application developers of both commercial and corporate in-house software benefit from a CTI API that provides telephony network- and hardware-independence. Rather than having to write separate applications for proprietary APIs, developers can write to a standard API, allowing one application to be used across a broad array of different types of networks and equipment. And because of the popularity of both Windows 95 and Windows NT, developers are offered the critical market mass to warrant an investment in the development of full-featured telephony applications. And, as has been noted, freeing developers from being tied to a particular hardware platform means they can focus on the primary task of creating innovative, easy-to-use applications, and not waste time having to worry about the plumbing.

Developers can choose the features they want to incorporate, and then link them together with a value-added interface consistent with the rest of the application. For example, if drag-and-drop is used extensively, a developer could let the user send files or facsimiles through the telephone to a colleague by dragging the icon of the file to an icon representing the colleague's destination. Similarly, the user could initiate conference calls by dragging three or four names from an electronic directory into a "Conference box" and clicking a "Connect" command. The developer can create the interface, and let TAPI carry out the work needed to make and manage the telephone connections.

Benefiting Hardware Manufacturers

Telephony network, switch, and hardware manufacturers benefit because there are more applications which can run with their products. They can focus on the area of their expertise—hardware, switching, network integrate—without needing to come up to speed on programming of user interfaces, database access, or other high-level tasks. A related benefit is that the open platform of TAPI 2.0 and Windows NT will mean that a full-range of broad- and vertical-market applications will emerge from third parties, greatly expanding the market for telephony hardware for PC platforms and PCs with integrated telephony capabilities—which means switch vendors will sell more hardware.

Benefiting Resellers

TAPI provides a great opportunity for resellers. The traditional telephony equipment resellers, who are selling PBXs and key systems, should see this as an opportunity

to capitalize on their expertise, by applying it to the new world of client/server telephony. They can then profit from the substantial new business that can be found in selling client/server telephony solutions. Computer resellers, and those providing LAN networking solutions, will find that computer telephony enabled by the Windows telephony platform provides a natural extension of their existing businesses.

An additional benefit to knowledgeable resellers is that providing computer telephony integration (CTI) solutions, is a robust market, where opportunities abound for resellers to bring value-added solutions to their customers. There is a large opportunity for resellers who can combine expertise in the field of desktop and server computing with a solid knowledge of traditional telephony. There are also opportunities for businesses with computer expertise to join with those having telecom expertise to create new products and services.

Benefiting Corporate Information Service and Telecom Managers

Corporate Information Service, network, and telecom managers will benefit because TAPI 2.0 and the Windows platform will open the door to a new world of telephony applications. Managers will see greater choices and benefit from increased vendor competition as the hardware abstraction of TAPI knocks down the old barriers created by stand-alone proprietary systems.

In addition to what should be lower purchase prices, managers will also benefit from lower costs of ownership from not having to manage a collection of disparate applications, tools, and operating systems. TAPI-enabled applications will also give managers the ability to provide new features for their customer population within their organization. And the convergence of Internet and PSTN technology should allow network managers to make more efficient use of existing network facilities.

Benefiting Call Center Managers

Call center managers benefit in much the same way as IS and telecom managers—with more competition and lower costs. The purpose of a call center is to provide rapid, high-quality customer service, and good tools let managers stay right on top of events, determine the nature of calls, and monitor agent availability. Historically, call center applications were only available from a limited number of vendors, and required mainframe or minicomputer class hardware support. They were also difficult to customize as the needs of the business changed, such as for short term promotions. Part of the advantage of client/server technology is that it enables the power, simplicity, and choice associated with of networked computing to be exploited. Thanks to applications such as Microsoft SNA Server, a move to client/server computing allows a company to continue to get value from its legacy mainframe and mini computers.

Benefiting Business Users and Consumers

Users will no longer be locked into a particular vendor or a single application. Just because a company, for example, has a particular PBX, it no longer means it has to use the single proprietary application. Competition in telephony application software means that a wide variety of applications will be available at competitive prices, and

new features will be continually rolled-out by commercial developers to meet growing user needs. This means end user investments in applications software and telephony hardware are protected by the interchangability of components.

TAPI provides users with easier-to-use applications involving telephone systems, particularly through the use of a graphical user interface on a user's PC. Performing functions such as conference, transfer, and hold, will become more intuitive. Also, new features will become available to users which they didn't have access to before. One example would be the integration of telephony with existing applications they may already be using, such as databases. And new applications will make it easier for users to manage their phone system, and to control how they are contacted, and by which means.

Some of the new applications will enable users to create different levels of caller accessibility to themselves. This will be based on incoming calling line identification, or other preferences. A user might want certain calls routed to them, such as from a spouse or children, no matter where the user is located. It will track that user down, even if they are out in the field and can only be reached on their cellular phone, for example. Similarly, they may want some callers to be sent to voice mail, while key customers might have access to their beeper. So TAPI-enabled applications will help users manage their time better, by putting them in control of their telecommunications.

Benefiting Public Network Operators

For public network operators, such as telephone companies and Internet service providers, TAPI-based applications and client/server telephony provide many of the same benefits as those enjoyed by application developers, such as the use of common tools. This greatly accelerates their ability to offer new services, which can either be provided by software and hardware developed internally, or purchased from a broad spectrum of vendors, thereby resulting in shorter time to market. TAPI-based applications also allow telephone companies and Internet service providers to provide value-added outsourced services for their customers, who will no longer have to purchase their own equipment. For companies that sell solutions, TAPI-based applications expand the range of products for them to sell to their customers.

SUMMARY

TAPI 2.0 and the Windows NT and Windows 95 operating systems provide the most powerful and flexible platforms for the development and use of CTI applications. TAPI 2.0 is supported by a full range of complementary APIs to enable a broad range of powerful, easy-to-use telephony and communications applications for a wide range of customers. Microsoft ActiveX Controls give developers powerful and easy-to-implement "plug-in" software components that simplify the development process, allowing developers to focus on creating their own value-added applications.

For developers and vendors of CTI software and hardware, the combination of Windows and TAPI 2.0 creates a new world of opportunities. As the computer and telephone are merged to create a new generation of communicating PCs, there should be an explosive demand for telephony applications, and the Windows NT and Windows 95 operating systems represent huge markets for such products.

And because TAPI 2.0 is built into Windows NT 4.0 (and will soon be made available for Windows 95), it will already be on a user's system, ready to enable whatever TAPI 2.0 telephony applications are placed on the market.

TAPI's hardware abstraction frees developers from having to create separate applications for each switch, just as TAPI's UniModem support protects developers from the exhaustive chore of having to directly support the vast array of modems.

CTI is beginning to revolutionize the manner in which people interact with telecommunications. This compelling combination of openness, comprehensiveness, scalability, and integration makes TAPI and Windows-based telephony the platform of choice for what should be an exciting and booming market.

MARKET SUPPORT

The number of TAPI-compatible products on the market continues to grow, supporting a broad range of telephone networks and applications. There is a wide spectrum of applications developers, both established companies and start-ups, coming from both the PC and telecommunications heritage. Every major PBX vendor in the world is shipping or has under development TAPI drivers for their systems. In addition to the PBX, drivers are available for the analog PSTN, ISDN, Centrex, client/server telephony, and even cellular.

Literally hundreds of thousands of developers worldwide have the ability to write TAPI solutions and they can receive the TAPI software developers toolkit and related documentation via their subscriptions to the Microsoft Developer Network. Because Microsoft does not require that each developer register its TAPI compatible product, it is difficult to gauge the precise number of TAPI products available. That being said, there are a some product listings and directories available in print, via the Internet, and referenced in market studies which capture at least some of the momentum and solution activity. You can find links to some of the companies offering TAPI-based products under the telephony solutions area on the Microsoft Web site for network communications and telephony at http://www.microsoft.com/communications.

FOR MORE INFORMATION

Customers

Contact your Microsoft representative, see your local reseller, or visit the Microsoft Internet site at http://www.microsoft.com.

For the latest in Windows network communications and telephony technologies and solutions, go straight to http://www.microsoft.com/ntserver/communications.

Developers

- Microsoft Developer Network CD-ROM: this quarterly subscription service includes all Microsoft software development kits and is a must for any developer using Windows. Call 1-800-759-5474 to order (if outside of the U.S., call +31.10.258.8864 in Europe or call +1-402-691-0173).
- Windows 95 Software Development Kit includes the TAPI SDK
- Internet: ftp.microsoft.com\developr\tapi

GLOSSARY

Here are quick definitions for some of the telephony words and terms used in this paper.

- **ActiveX Controls**—A set of technologies that enable software components to interoperate in a networked environment, providing end users with a richer, more interactive experience. The ActiveX technologies are enhancements to OLE, Microsoft's component software technology. ActiveX Controls are small, efficient modules that implement specific, specialized functions.
- **API**—An application programming interface is a standardized set of procedure calls that can be used to interface applications with telecommunications protocols.
- **Auto attendant**—"If you know the extension of your party, enter it now. If you would like to speak to an operator, press two." Auto attendants are being used more and more by businesses large and small to handle initial routing of incoming telephone calls.
- **Call center**—A place with agents or operators at banks of telephones to either make outgoing, or field incoming, telephone calls. For example, a bank or credit card company uses call centers for inquiring about overdue accounts, and many corporations have help desk operations to handle incoming customer queries. Call center often has automated dialing, predictive dialing, one-step forwarding, and other telephony applications.
- **CTI**—Computer-telephone integration. The merging of the computer and telephone, which will transform the personal computer from being an information processing device to also being a powerful platform for communications. Linking telecommunications to the ever-increasing processing capabilities and rich user interface of the computer will enable new forms of communications and richer access to existing types of communication, including voice, asynchronous data, fax, remote access to LANs, Internet access, on-line services and more. "The Communicating PC" will redefine how we share ideas and information, and provide a portal to other people, computers, and network services, anywhere in the world.
- **Integrated (or *unified)* messaging**—Also called unified messaging, is the concept of unifying all incoming messages onto a single computer screen. A telephony application that offered unified messaging would present a user with a single list showing all new electronic mail, telephone messages, and faxes. Clicking on a telephone message could result in either the telephone ringing with the message, or with the computer's sound card relaying the message. Unified messaging gives users the luxury of scanning their messages, and deciding which to deal with first.
- **IP—**Internet Protocol, the critical Internet standard governing connectionless mode network services.
- **ISDN**—Integrated Services Digital Network, AT&T's international standard for voice, data, and signaling via end-to-end digital circuits and providing significantly enhanced transmission speeds.
- **IVR**—Interactive voice response, used in telephony applications in which a user interacts with a computer by pressing TouchTone keys in response to a set of recorded choices. IVR systems are used for creating automated banking statement services, providing train and movie schedules, and such. IVR systems can also use a text-to-voice synthesizer to increase its repertoire of responses.

- **MAPI**—The Microsoft Windows Messaging Application Programming Interface, a set of API functions that enable messaging. Part of the Windows Open Services Architecture (WOSA).
- **PBX**—Private Branch eXchange, the private switching network some companies, especially larger ones, use to route their corporate telecommunications.
- **Personal call manager**—Brings the management of telephone calls onto the computer screen where users can initiate a phone call with the click of the mouse, choose to respond to calls via Caller ID, selectively forward calls to a conference room or other site, and in other ways personalize their response to telecommunications.
- **PIM**—Personal information managers are like computerized appointment books, and are used on many personal computers, to help a person keep track of appointments, telephone numbers, and other contact information. A telephony enabled PIM could provide point-and-click telephone dialing, automated conference calling, or automated access to databases and other applications.
- **Predictive dialer**—Used in call centers to place outgoing telephone calls faster than could be done by the center's personnel. Predictive dialing is based upon the fact that many outgoing calls will either reach busy signals or go unanswered, something which used to waste the time of call center personnel. A predictive dialing system is able to detect a phone pickup and route the successful call to a call center person before the person on the other line has the chance to say "hello." This is used in outbound telemarketing operations, collection systems, and in other applications.
- **PSTN**—Public Switched Telephone Network, the more formal term for the local telephone company.
- **SAPI**—The Windows Speech API, a WOSA element, provides services that enable the use of speech recognition and/or text-to-speech in applications.
- **TAPI**—The Windows Telephony API provides services that enable an application developer to add telephone communications to applications developed for operating systems that support the Microsoft Win32 application programming interface, such as Microsoft Windows NT and Microsoft Windows 95. The combination of TAPI, the Windows platform, additional communications-related Windows APIs, and ActiveX Controls provides the ideal platform for developing and using telephony applications.
- **UniModem**—Microsoft's universal modem driver greatly simplifies the development process by providing an abstraction layer to which programmers can make generic calls, while on the other side a vast collection of drivers are used to translate the abstracted command into a specific action on whatever modem the operating system has detected.
- **Visual Call Control**—A type of telephony application that allows users to manage their telecommunications from their computer screen. For example, with such an application a fax could be sent to another person by dragging an icon of the fax across the screen and dropping it on the name of the recipient. The telephony application would work behind the scenes to look up the persons telephone number, dial it, establish a communications link, and transmit the fax. Similarly, a conference call could be arranged by clicking and dragging the names of four individuals

onto a conference call icon.

- **WAV**—Wave Audio, the Win32 function for handling sound.