# CT-Connect

## Introduction

Order Number:   05-0414-004

**Revision Information:**       This manual is an update

**Software/Version:**          CT-Connect™ Server Version 3.0
                               CT-Connect Application Programming
                               Interface Version 3.0

# Contents

## Index

## Figures

## Tables

# About This Manual

This manual provides an overview of Dialogic®'s CT-Connect (CTC), computer-telephony concepts, and telephony application functions.

## Audience

This manual is for anyone interested in CTC.

## Associated Documentation

### CT-Connect Documentation

For more information about CTC, refer to the following:

- *CT-Connect Installation and Administration Guide* — This manual describes how to install and manage the CTC Server software on a server system and the CTC API software on client systems.

- *CT-Connect Programming Guide* — This manual provides detailed descriptions of the CTC API routines and guidelines for using these routines. It also includes details of the operational differences for links to specific switches.

- *CT-Connect Release Notes* — These online notes provide information about changes to the CT-Connect software and/or its documentation at the time of release.

### Dialogic Web Site

For more information about CT-Connect, and other Dialogic products, visit Dialogic's web site at HTTP://WWW.DIALOGIC.COM.

### Switch Documentation

Refer to the documentation supplied with the switch for details of its features and characteristics.

## Terms and Definitions

The following terms are used throughout this manual:

| Term | Definition |
| --- | --- |
| OpenVMS | Refers to the OpenVMS™ VAX™ and OpenVMS Alpha operating systems. |
| Windows 3.1/3.11 | Refers to Microsoft® Windows™ 3.1, Windows 3.11, and Windows for Workgroups 3.11 |
| CTC server | A supported system running the CTC Server software. |
| CTC client | A supported system running the CTC API software |
| Communications Link | The logical link between the CTC server and the switch. |
| Switch | The telephony switching device. For example, a Private Branch Exchange (PBX), Private Automatic Branch Exchange (PABX), or central office switch. |

# 1

# Overview

## 1.1  What is Computer/Telephony Integration?

Computer/Telephony Integration (CTI) is the combination of telephone and computer technology. This combination enables voice and data processing equipment to work together so that people can exchange information more quickly, easily, and efficiently.

## 1.2  What is CT-Connect?

CT-Connect is a software toolkit for developing CTI applications. It enables you to create applications that monitor and control calls, whether they are intended for large, call-center environments or for use on individual PC desktops.

You can write a CTC application to perform standard telephony activities such as making, transferring, and receiving calls. Your CTC application can add value to your telephone system by providing features such as retrieving data relevant to a call or displaying information on callers before their call is answered.

For example, an agent who takes customer orders by telephone can use a CTC application to:

- Collect data from the caller, for example, a customer reference number

- Display details of the customer's account

This can be done without any input from the agent; the customer uses their telephone keypad to enter their reference number, the CTC application uses these digits to look up the customer's account details and display them automatically on the agent's desktop.

Queries about outstanding orders can now be answered quickly, and new orders placed without retaking known customer details.

## 1.3   CT-Connect in a Call-Center Environment

CTC is an important addition to any computer/telephone environment. In a call center, it can be used to automate call processing, data retrieval, and information distribution.

A call center is a group within an organization that does its business primarily by telephone. Its function can range from telemarketing to customer services, from order processing to field service dispatch. However, it can be defined as a center where a large number of agents or operators receive and respond to caller requests.

Users in a call center may make use of a wide variety of telephone and computer equipment, including:

- Telephones (headsets or hand-held sets) connected to telephone switches

- Switches that provide call handling for the call center environment, for example, Automatic Call Distributors (ACDs)

- Corporate databases (for example, customer databases, financial databases)

- Computer networks (both Local Area Networks (LANs) and Wide Area Networks (WANs))

- Fax machines

- Voice messaging systems

- Voice Response Units (VRUs)

- Workflow management systems

CTC can make call-center voice and data processing equipment work together effectively and efficiently. Section 1.5 describes some of the features you can build into your CTC application for the call-center environment.

## 1.4   CT-Connect on the Desktop

A CTC application designed for desktop use can also combine essential telephony and system functionality. Although desktop users have different requirements to call-center users (for example, responding to callers individually), they may still need to integrate telephony functionality into tailored applications.

CTC enables you to:

- Write desktop applications that perform a range of telephony features and provide access to data processing equipment. Section 1.5 describes some of the features you can build into any CTC application.

- Create or use existing Microsoft's Windows 3.1/3.11 TAPI applications, without installing proprietary telephony hardware on a PC desktop. Section 2.10.1 describes how CTC provides support for TAPI.

- Modify familiar PC desktop applications that use Dynamic Data Exchange (DDE). This enables you to add  telephony functionality to familiar applications such as Microsoft Access® or Excel®. Section 2.10.2 describes how CTC provides support for DDE.

- Allow TSAPI applications on a Windows 3.1/3.11 CTC client system to access the telephony features of a switch through the CTC server. Section 2.10.3 describes how CTC provides support for TSAPI.

- Use an ActiveX® control to access the CTC API from an ActiveX container. The ActiveX control is supported on Windows 95 and Windows NT systems configured as CTC clients. Section 2.10.4 describes how CTC provides support for ActiveX controls.

- Run "off-the-shelf" telephony applications; CTC can be used as the interface to a variety of switches.

## 1.5   CTC Telephony Functions

The following sections describe some of the telephony functions CTC provides and how they can be used by a CTC application.

### 1.5.1   Call Control

Using computer-telephony applications, telephones can be controlled either individually, for example, from a PC, or remotely by the switch. Generally, how the telephone is controlled determines the functionality that your application can provide.

**First-Party Call Control**

First-party call control means that the application controls the call over a direct connection between the system and the telephone (or other telephony device).

Typically, a first-party call-control application would be used by an individual to access a limited set of telephony functions from their desktop. It usually requires a physical connection to a telephone through a board installed on the PC, and a different board may be required for each type of telephone.

**Third-Party Call Control**

Third-party call control means that the application controls a call by communicating with the switch across a network connection. No direct connection between the application system and the telephone is required.

CTC enables your application to use third-party call control by providing access to the switch over a communications link (through a data communications port on the switch). Because the call is controlled at the switch and not limited to controlling one telephone, your application can provide additional features. For example, it could automate dialing for a group (see Section 1.5.8) or enable call routing (see Section 1.5.7).

Chapter 2 provides more information about the CTC software and how it provides access to the switch.

## 1.5.2 Calling Line Identification

Calling Line Identification (CLID) identifies the telephone number or extension number from which the call originated. Even before a call is answered, database lookups can be made and information displayed on the desktop system of the agent whose telephone receives the call. CLID significantly reduces the time the agent spends to get customer details on the screen, increasing customer satisfaction and reducing the duration of calls.

Some switches provide Automatic Number Identification (ANI) instead of CLID. ANI is similar to CLID but, depending on the configuration of the originating switch, ANI can be used to identify either of the following:

• The telephone number from which the call originated

• The number for the switch from which the call originated (for example, the external telephone number for a building rather than an extension number)

## 1.5.3 Dialed Number Identification Service

Dialed Number Identification Service (DNIS), also known as Dialed Digits, identifies the telephone number that the caller dialed. For example, customers use different telephone numbers for placing orders or reporting a fault. Although all of these calls may go to a central queue, the CTC application can be set up to identify the dialed number and use this to correctly display either the order entry or field service form on the agent's screen.

### 1.5.4  Voice Response Integration

A Voice Response Unit (VRU), also known as Interactive Voice Response (IVR), presents callers with a voice menu of customer options. Callers can select an option either by speaking or by pressing touch-tone digits. These devices can enable customers to access and manipulate database information. For example, a customer can obtain the balance of their current account or transfer money from one account to another.

After being connected to an agent or VRU, the caller may need to be transferred. With CTC, the call and the in-process transaction information can be automatically, and simultaneously, transferred to the appropriate desktop.

### 1.5.5  Call/Data Transfer

When a telephone call is transferred, any details taken by the agent can be transferred with the customer data that has already been retrieved. For example, if a customer places an order and then wants to be transferred to the accounts payable department, the customer's order information can be transferred too. This avoids customer frustration when different departments each have to perform data entry.

### 1.5.6  Voice Mail

Voice mail enables a caller to leave a message with the appropriate person at any time of day. Typically, the caller is played a personalized voice greeting and presented with a menu of options.

To select an option, the caller enters digits (as with a VRU system). These digits can be used to identify which agent or group they want to contact and the relevant voice mail account. Any message that they leave in an agent's voice mail box can be played back by that agent over the telephone.

Depending on the switch, a CTC application can control the greeting played in response to the caller, collect digits entered by the caller, and transfer the call to another number.

### 1.5.7  Call Routing

Call routing can be used in a variety of ways. For example, you can identify important customers and prioritize their calls. You may have categorized your customers into Platinum, Gold, and Silver groups. Using call routing, you can make sure that calls from your Platinum customers (identified by ANI or a VRU) have a minimum wait time.

Another example of call routing is where you identify customers with overdue accounts. Having identified a caller with a negative balance on their account (through ANI information or a VRU), the call can be routed to the collections department, no matter which department the caller originally dialed.

Call routing can also be used to ensure that a customer is always connected to the same agent, giving improved customer service.

### 1.5.8  Auto Dialing

CTC applications can be used to automate dialing functions through a variety of techniques, including:

- Preview dialing

  The agent instructs the CTC application to dial a number after viewing the customer data.

  When the call has been completed, or the call is not successful (that is, the customer's telephone is engaged or there is no answer), the application displays the data for the next customer and dials that customer's number.

- Predictive dialing

  The CTC application handles high-volume call processing. It dials a string of numbers for a group or pool of agents. When the calls are answered, CTC transfers them and their associated data to available agents.

  An algorithm can be used to pace predictive dialing. This can be based on various factors, for example, the number of agents available or the number of expected failed calls.

### 1.5.9  Management Reporting

Information provided by both the telephone system (number of calls, busy hours, agent statistics, queue time, and so on) and the computer system (how many orders, who ordered, what was ordered, and so on) can be merged in a single comprehensive report. This report can be used to help evaluate agent and product performance, track results of marketing campaigns, and identify new selling opportunities.

## 1.6  An Example CTC Application

CTC uses the features and functions of your telephone system and accesses the information, distribution, and management capabilities of your computing environment.

The following paragraphs describe an example CTC application for an order entry call center.

Based on information obtained from ANI or DNIS, the CTC call center application routes incoming calls from the voice switch to a VRU, a voice messaging system, or an agent's desk. If ANI and/or DNIS are not in an application, information may be obtained by routing calls to the VRU and requesting information such as a customer account number.
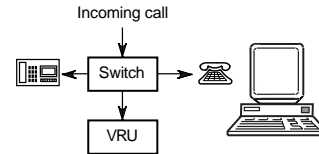


When a call is routed to an agent's desk, the CTC application displays customer records and product or service scripting information on their desktop systems. Call-related information stored at various databases across the network is transferred to the agent's desktop as calls are received. This can include information such as account profiles, stored voice, document images, and so on. Screen prompts, or scripts, lead agents through appropriate responses and flag cross-selling opportunities.
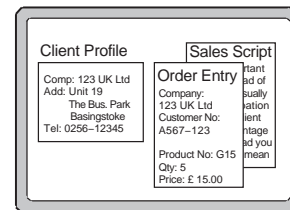


When an agent needs to consult with a supervisor or transfer a call to another department, the CTC application sends the call, and all related data, to the appropriate staff member.
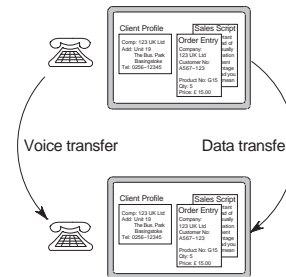


Agents update customer records based on the outcome of call transactions, then send the updated information over the network to the relevant databases. Information on orders is forwarded over the network for fulfillment by appropriate departments.)
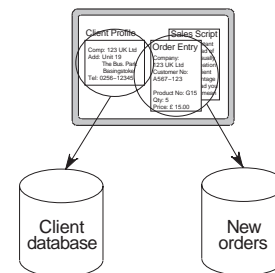
Completed order forms are sent from agents' desktop systems to fax devices that forward hardcopy confirmations to customers.
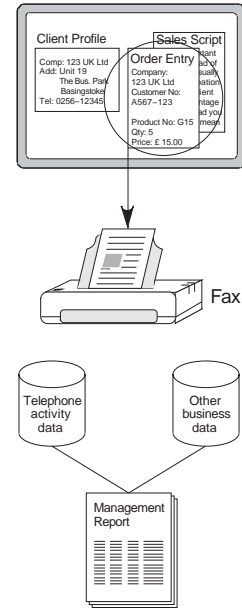
Statistical reports on call center activity are compiled by merging data on telephone activity supplied by CTC with information collected from databases across the network and workflow information generated by the call center application.

# 2

## The CT-Connect Environment

CTC is based on a client/server architecture. This chapter provides an overview of this architecture and describes examples of CTC in environments with client/server topologies. It also provides details of the platforms, switches, and protocols that CTC supports.

## 2.1 Client/Server Architecture

Client/server computing uses the best aspects of traditional styles of computing—mainframe production systems, multiuser departmental systems, and standalone PCs or workstations—and creates an efficient, distributed work environment.

The client/server principle is based on distributing individual tasks to servers on the network. For example, the data management task is executed by a database server which fulfills the needs of multiple clients.

Clients communicate with servers by making "requests". The server typically returns a "response" on fulfilling the request. The response may be the return of information requested by the client, and/or an acknowledgement that the request has been carried out, whether successfully or not. This way, computing becomes distributed by sharing both data and processing power among the computing resources available. Through servers, users on client systems are given access to scarce resources such as high-speed printers, or, as is the case with a CTC server, a telephone switch.

Typically, tasks that are supported by servers are:

- Data management
- Print functions
- Fax functions
- Imaging functions

A client/server architecture also means that different hardware and software, operating systems, and so on, can be used in the network.

### 2.1.1  CTC Client/Server Software

The CTC software comprises:

- CTC Server

  The CTC Server software receives telephony requests from CTC clients, converts these requests to a switch-compatible protocol and then forwards them to the switch. When the switch responds, the CTC Server software returns the results to the clients. It provides the single point of contact between one or more CTC clients and the switch.

  In addition, the CTC Server software detects telephony events and distributes information to any CTC clients requesting it.

- CTC Application Programming Interface (API)

  This is the CTC client. It consists of a set of CTC routines that are used by CTC applications to request telephony tasks. It interfaces between a CTC server (a PC running the CTC Server software) and the CTC application.

Since CTC is based on a client/server architecture, it is flexible; it works in a fully implemented client/server environment, and with existing legacy systems, for example, mainframes.

The following sections provide examples of CTC in different system environments. Section 2.2 describes CTC in a fully implemented client/server environment. Section 2.3 describes CTC in a traditional environment.

## 2.2  CTC in a Fully Implemented Client/Server Environment

Figure 2–1 shows CTC in an example of a *fully implemented client/server environment*:

- Data and file management, printing and faxing tasks have been distributed onto servers.

- Users each have access to PCs or workstations.

**Figure 2–1   CTC in a Fully Implemented Client/Server Environment**
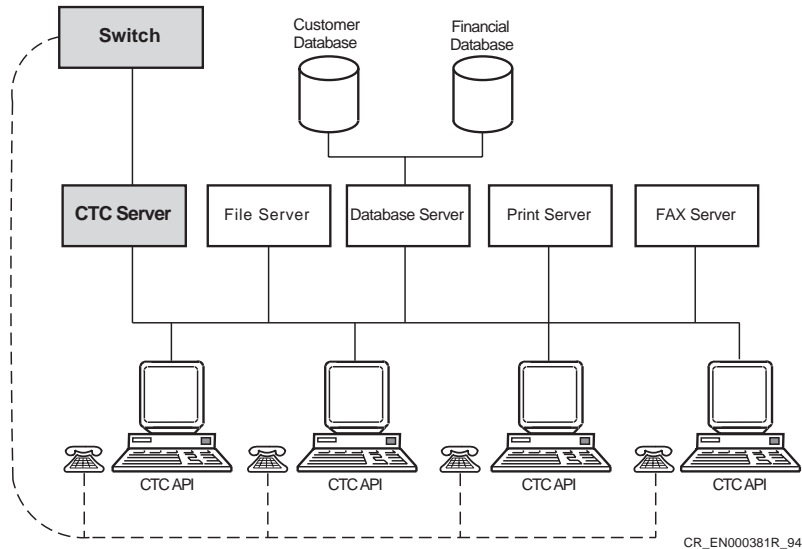


CR_EN000381R_94

In this example, CTC supports the environment in the following way:

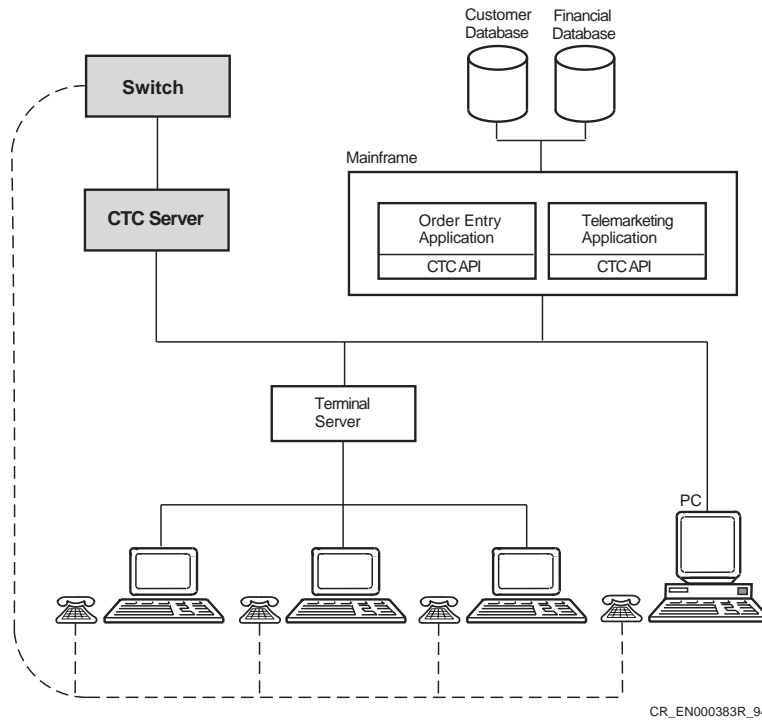- The CTC server provides the link between the telephone switch and the computer network.

- Each PC, or workstation, has a copy of the CTC API installed.

- A CTC application is running on each user's PC or workstation.

- The CTC application uses existing database structures. It accesses the database server, file server, and existing printing and faxing facilities.

## 2.3   CTC in a Traditional Environment

Figure 2–2 shows CTC in an example of a *traditional environment*:

- A mainframe and/or minicomputers support the legacy systems.

- The mainframe acts as a type of master server, mainly to provide access to databases and peripherals.

- Processing is mainly done on the mainframe.

- Most of the users still use terminals, although some PCs and/or workstations may be present.

**Figure 2–2   CTC in a Traditional Environment**



CR_EN000383R_94

In this example, CTC supports the environment in the following way:

- The CTC server provides the link between the telephone switch and the computer network.

- The CTC API has been added onto each of the existing applications that are going to use CTC functions.

- These modified applications use existing database structures accessed through the mainframe.

- The PC accesses the applications through terminal emulation.

## 2.4  Combined Environments

CTC's flexibility means that it can run in an environment that includes aspects of both the fully implemented client/server environment and the traditional environment. For example, the CTC API can be installed on each PC as in Figure 2–1 but data retrieval may be performed using legacy applications as in Figure 2–2.

## 2.5  Supported Platforms

Table 2–1 shows the platforms supported by the CTC software. For details of which versions are supported, refer to the *CT-Connect Installation and Administration Guide*.

**Table 2–1  CTC Supported Platforms**

| CTC Software | Supported Platform |
| --- | --- |
| CTC Server | Windows NT Workstation or Windows NT Server |
| | Sun Solaris |
| CTC API | Digital UNIX |
| | HP-UX |
| | OpenVMS Alpha |
| | OpenVMS VAX |
| | OS/2 Warp |
| | SCO UnixWare |
| | SCO OpenServer |
| | Sun Solaris |
| | Windows 3.1/3.11 |
| | Windows 95 |
| | Windows NT Workstation or Windows NT Server |

## 2.6  Supported Switches

CTC supports applications linked to a large number of popular switches, implementing both industry standard and proprietary protocols:

- CTC implements the industry standard Computer Supported Telecommunications Applications (CSTA) Phase I and Phase II protocols. CSTA protocols are defined by ECMA, a standards body for information and communication systems. The implementation of CSTA allows any Telephone Equipment Manufacturer (TEM) to provide a link to CTC from a CSTA-compatible switch.

  Example switches supported by CTC using the CSTA protocols are:

  – Ericsson® MD110 (CSTA Phase I)

  – Alcatel 4400 (CSTA Phase I and Phase II)

  For a full list of supported switches, contact Dialogic.

- Switches supported by CTC using proprietary link protocols are:

  – Lucent DEFINITY® G3 (ASAI protocol)

  – Nortel™ Meridian 1™ (Meridian Link Protocol)

## 2.7  Supported Network Protocols

Table 2–2 lists the supported network protocols for the data network linking CTC clients with the CTC server.

**Table 2–2  Supported Network Protocols**

| Client Operating System | Sun Solaris Server Network Protocol | Windows NT Server Network Protocol |
|---|---|---|
| Digital UNIX | TCP/IP | DECnet† |
| | | TCP/IP |
| HP-UX | TCP/IP | TCP/IP |
| OpenVMS | TCP/IP | DECnet |
| | | TCP/IP |
| OS/2 Warp | TCP/IP | TCP/IP |
| SCO OpenServer | TCP/IP | TCP/IP |
| SCO UnixWare | TCP/IP | TCP/IP |
| Sun Solaris | TCP/IP | TCP/IP |
| Windows 3.1/3.11 | TCP/IP | TCP/IP |
| Windows 95 | TCP/IP | Named Pipes |
| | | NetBIOS™ over NetBEUI |
| | | NetBIOS over TCP/IP |
| | | Novell® SPX |
| | | TCP/IP |
| Windows NT | TCP/IP | DECnet |
| | | Local RPC |
| | | Named Pipes |
| | | NetBIOS over NetBEUI |
| | | NetBIOS over TCP/IP |
| | | Novell SPX |
| | | TCP/IP |

†DECnet™/OSI must be installed on Digital UNIX clients for DECnet Phase IV compatibility.

## 2.8   Additional DCE/RPC Software Required

Some of the CTC client platforms require that you install additional software to include the Distributed Computing Environment (DCE) and Remote Procedure Call (RPC) services. CTC requires these services to operate and to support application development but in some cases they are not provided by, or included in, the base operating system.

Table 2–3 lists the CTC client platforms and indicates if you need to add DCE/RPC software in each case. Please refer to the *CT-Connect Installation and Administration Guide* for details of software product names and versions.

**Table 2–3   Additional DCE/RPC Software Required**

| Client Operating System | Additional Software Needed |
| --- | --- |
| Digital UNIX | Yes |
| HP-UX | Yes |
| OpenVMS | Yes |
| OS/2 Warp | Yes |
| SCO OpenServer | Yes |
| SCO UnixWare | Yes |
| Sun Solaris | Yes |
| Windows 3.1/3.11 | No |
| Windows 95 | No |
| Windows NT | No |

## 2.9   Portability

Most of the functions from the CTC API are supported by the telephone switches and links described in Section 2.6. If your application is programmed to use these common functions, it can be ported across different telephone switches.

In addition, the common CTC API can be ported across the supported computer platforms. For more information and details of the CTC API programming routines, refer to the *CT-Connect Programming Guide*.

## 2.10 CT-Connect Additional Interfaces

CTC supports additional interfaces to the following:

- Windows Telephony Application Programming Interface (TAPI) (see Section 2.10.1)

- Windows Dynamic Data Exchange (DDE) (see Section 2.10.2)

- Novell Telephony Services Application Programming Interface (TSAPI) (see Section 2.10.3 )

- Microsoft ActiveX controls (see Section 2.10.4)

These additional interfaces are not included with the CTC software and you should contact Dialogic if you want to obtain them. The Dialogic web site provides more information on these interfaces (the web site location is http://www.dialogic.com).

### 2.10.1 CT-Connect and Microsoft TAPI

CTC provides a Service Provider Interface (SPI) to support Microsoft's TAPI. The CTC SPI supports the Windows TAPI first-party call-control functionality, enabling applications to use end-point telephony functions.

A Windows TAPI application requires two interfaces:

- TAPI, which contains the functions that the application uses to perform telephony operations.

- An SPI, which enables the application to access the physical device, effectively, a device driver. The SPI is not provided as part of TAPI but must be supplied by a third-party manufacturer or developer.

The TAPI SPI provided by CTC enables applications to access the CTC API and the communications link between the CTC server and a switch. This means that the TAPI application can use first-party call control without the need for one or more proprietary boards installed on the PC.

CTC supports most of the functions provided by the following TAPI services:

- Assisted Telephony Services

- Line Device Services

For more information on the CTC TAPI interface, contact Dialogic.

### 2.10.2 CT-Connect and DDE

CTC also supports Dynamic Data Exchange (DDE).

DDE is a protocol that enables Windows applications to "talk" to each other. Windows applications can send data or receive data from other Windows applications running on the same system, providing they support compatible DDE interfaces. Sending and receiving data between these applications is continuous and automatic: as soon as data is received by one application, the other is notified.

Combined with the capabilities of DDE, CTC provides you with a powerful tool you can use to transform desktop applications by adding essential telephony functionality. Using the CTC DDE interface, familiar Windows applications can transmit and receive telephony information. For example, you can add an "Answer call" button on a Microsoft Access form. When you click on the button, Microsoft Access and the CTC DDE interface exchange information so that the call is answered and the appropriate data base record is displayed.

From Excel you can use the CTC DDE interface to automatically place a call as soon as an item on a spreadsheet changes, or, display a relevant Excel spreadsheet as soon as a call comes in.

The CTC DDE interface is designed to work in the desktop environment. It supports one logical channel between a process on your PC and a telephony device (typically the telephone next to your PC).

For more information on the CTC DDE interface, contact Dialogic.

### 2.10.3 CT-Connect and Novell TSAPI

The CTC TSAPI interface enables TSAPI applications on a Windows 3.1/3.11 CTC client system to access the telephony features of a switch through the CTC server.

The CTC TSAPI interface supports a subset of TSAPI functions and services.

For more information on the CTC TSAPI interface, contact Dialogic.

For more information about TSAPI itself, refer to Novell's *Netware Telephony Services Release 2.21 TSAPI* document.

### 2.10.4 CT-Connect and Microsoft ActiveX Controls

The CTC ActiveX control provides access to the basic functions and features of the CTC API. The ActiveX control is supported on Windows 95 and Windows NT systems configured as CTC clients.

The control can be used with any ActiveX containers, including applications such as Microsoft Access® or programming environments such as Visual Basic.

For more information on the CTC ActiveX control, contact Dialogic.

For more information about using ActiveX, refer to your ActiveX container documentation (for example, the Visual Basic documentation) or see Microsoft's web page: http://www.microsoft.com.

# Index