

2 Introduction

The Novell® NetWare® Network Operating System (NOS) provides APIs that allow client and server applications to offer users a variety of computing services (for example, file and print services). NetWare Telephony Services expand the set of NetWare services and bring together the two most common pieces of equipment on an end user's desktop, the telephone and personal computer. AT&T and Novell integrated these two environments using TSAPI, a switch-independent standards-based API.

Purpose

This document specifies Telephony Services Application Programming Interface (TSAPI) services and C programming language syntax. TSAPI presently supports Microsoft® Windows™ client, Novell® NetWare® client, and Novell® NetWare® server environments. The architecture supports future growth into other client and server environments while preserving the TSAPI programming interface. Any operating system specific details are marked as such in the TSAPI specification.

The reader should be familiar with telephony and the ECMA Computer-Supported Telecommunications Application (CSTA) service, and protocol definitions, i.e., ECMA-179 and ECMA-180. The ECMA CSTA standard is the basis for TSAPI service and parameter definition. Write to ECMA at the address below to obtain copies of these standards.

European Computer Manufactures Association
114 Rue du Rhône
CH-1204 Geneva
Switzerland
Telephone - 41 22 735 36 34

μ §"41" above is the United States International Calling code for Switzerland.

Product ArchitectureXE "Architecture"§

NetWare XE "NetWare NOS"§ is a distributed client application environment where multiple processors (client and servers) can interact with each other using the NetWare NOS. This computing environment, along with NetWare Telephony Services, supports standalone or distributed telephony applicationsXE "Applications"§ on multiple clients/servers. In the case of a distributed application, client and server platforms contain the appropriate application components. For example, the client platform may embody the user interface and local control of the user's telephone. The server platform may use call information to route calls or look up caller information before the switch delivers calls to a users' telephones. NetWare supports the client/server communication and interaction between Telephony Services and the application (client or server based).

Since Telephony Services provides a standard API (TSAPI), it facilitates the development and use of telephony enabled client and server applications. Figure 2-1 illustrates the architecture of Telephony Services. Note that a switch-specific driver terminates the CTI link, thereby making the other Telephony Services modules switch independent.

Figure 2-1

Telephony Services Architecture

μ §

Telephony Services consists of the following hardware and software components (refer to Figure 2-1):

- ◆ **CTI Link** - this link supports the Computer Telephony Integration (CTI) protocol that logically integrates the telephone and the client workstation at users' desktops. A CTI link connects link specific hardware in the server and the switch. It provides sessions between TSAPI applications and the call processing software within the switch. The CTI link is PBX specific.
- ◆ **CTI Link Hardware** - is any server hardware that terminates the CTI link to a switch. This hardware is PBX vendor specific.
- ◆ **Switch Driver** - a set of software modules (NLMs) which support and terminate the switch specific CTI link and protocol; map the CTI protocol to the TSAPI (if required); support any Administration and Maintenance capabilities of the switch driver (if any); and support a driver interface to the CTI Link Hardware. The Switch Driver modules are PBX vendor specific.
- ◆ **Switch Driver Interface** - a software interface between the Switch Driver and the Telephony Services NLM that passes messages between applications and the switch driver. Typically, the messages

consist of CSTA service requests, responses, and events for TSAPI clients. The messages may also be administration and maintenance requests, responses, and events for an application doing PBX Driver Administration or Maintenance. The Switch Driver Interface is PBX independent **XE "Switch:Independent" §** and supports any Telephony Services compliant driver.

- ◆ **Telephony Services NLM (Telephony Services NLM)**
XE "Telephony Services NLM" § XE "Server:NLM:See Telephony Services NLM" \t " §- this software module provides communication between multiple telephony-enabled applications and the switch driver. The Telephony Services NLM routes messages from the Switch Driver to the applications waiting for telephony events and passes the messages received from applications (TSAPI Service Requests) to the Switch Driver. All messages between client applications and a PBX Switch Driver pass across the Switch Driver Interface **XE "Switch:Driver:Interface" §**. The Telephony Services NLM enforces user restrictions administered **XE "Administration" §** in the Telephony Server's security database. This module is switch independent **XE "Switch:Independent" §** and supports any Telephony Services compliant driver.
- ◆ **Telephony Server** - for convenience, an instance of the NetWare Telephony Services NLM software running on a particular server is referred to as a Telephony Server.
- ◆ **Telephony Services API (TSAPI)** - the CSTA-based, C language definition of the functions (services), data types (parameters and structures), and event messages that telephony-enabled applications use to access Telephony Services. This document specifies TSAPI. TSAPI is switch independent **XE "Switch:Independent" §** and supports any Telephony Services compliant driver.

- ◆ **Telephony Server Library**`XE "Server:Library"§` - server-based applications use this software module to access TSAPI functions. This library accepts TSAPI Service Requests and delivers responses and events to server applications. This library can run on the same physical server as Telephony Services or on any NetWare server in the network. This module is switch independent`XE "Switch:Independent"§` and supports any Telephony Services compliant driver.
- ◆ **Telephony Client Library**`XE "Client:Library"§` - client-based applications use this software module to access TSAPI functions. This library accepts TSAPI Service Requests and delivers responses and events to client applications. This module is switch independent`XE "Switch:Independent"§` and supports any Telephony Services compliant driver.

In summary, the following software modules and interfaces are switch-independent`XE "Switch:Independent"§` and support any Telephony Services compliant driver:

- ◆ Switch Driver Interface`XE "Switch:Driver:Interface"§`,
- ◆ Telephony Services NLM`XE "Telephony Services NLM"§`,
- ◆ Telephony Services API (TSAPI),
- ◆ Telephony Server Library`XE "Server:Library"§`, and the
- ◆ Telephony Client Library`XE "Client:Library"§`.

The following software/hardware modules are switch specific`XE "Switch:Specific"§`:

- ◆ CTI Link`XE "CTI:Link"§`
- ◆ CTI Link Hardware`XE "CTI:Link hardware"§`, and the
- ◆ Switch Driver`XE "Switch:Driver"§`.

Telephony Services Applications

XE "Applications" § Telephony Services supports a variety of telephony-enabled applications. It can support voice control applications that allow the user to manage and control incoming and outgoing calls at the desktop or more complex applications for the office work group or call center environment.

Applications can provide a variety of features to enhance user telephone control from the client workstation. Application features may include:

- ◆ call management,
- ◆ call screening,
- ◆ call logging,
- ◆ directory dialing from personal (client), workgroup (server), and corporate directories,
- ◆ dialing and integration with other applications, and
- ◆ integration of message waiting indicator for email or other messaging applications.

Telephony-enabled applications can meet customer needs in many markets. For example, a customer service application can allow agents to interact with both the "telephony" and "caller database" aspects of the job. An easy-to-use GUI application can include the caller information on a "screen pop" at agents' PCs. Any server on the network (PC, mini-computer, or mainframe) can contain a caller database. The application integrates access to all the different information (voice and data) needed to support an inbound customer service center. Another example of such an application is call routing. Here, the switch requests that the server provide a destination for incoming calls. Applications may also request outbound calls for Outbound Call Management (OCM) applications such as predictive dialing.