*chapter*

# 3 *TSAPI CALL MODEL*XE "Call:Model"§

This chapter describes concepts from the ECMA CSTAXE "CSTA"§ standard that are important for TSAPI applicationXE "Applications"§ programming. The information presented here is summarized from the CSTA specifications. The complete specification is available from ECMA at the address given in Chapter 2.

## Terminology

The following sections provide TSAPI definitions and acronyms. For clarity, the ECMA terms begin with capital letters throughout this section.

### Definitions

**ACD Agent**XE "ACD:Agent"§XE "ACD"§**:** A telephony user that is a member of an inbound or outbound Automatic Call DistributionXE "Automatic Call Distribution:See ACD" \t " "§ (ACD) group. ACD Agents first sign on (Login) to an ACD groups and then the ACD will distribute calls to the agent.

**Active CallXE "Call:Active"§:** The call (at a station) that is connected (in a talking state) at that station. More specifically, the Connection (see *Connection*) for the Active Call is in the Connected State (see *TSAPI Connections* section, *Connection State* definition).

**Alerting CallXE "Call:Alerting"§:** A call that is ringing at a Device. More specifically, the Connection (see *Connection*) for an Alerting Call is in the Alerting State. When the Device is a telephone, the Alerting Call is ringing the telephone instrument.

**ApplicationXE "Applications"§:** A co-operative process distributed between a Switching FunctionXE "Switching:Function"§ (see *Switching Function*) and a Computing FunctionXE "Computing:Function"§ (see *Computing Function*).

**Application DomainXE "Application domain"§:** The union of one Switching Sub-DomainXE "Switching:Sub-domain"§ (see *Switching Sub-Domain*) and one Computing Sub-DomainXE "Computing:Sub-domain"§ (see *Computing Sub-Domain*).

**Basic CallXE "Call:Basic"§:** A Call (see *Call*) between exactly two Devices (see *Device*).

**CallXE "Call"§ (TSAPI programming objectXE "TSAPI:Programming object:See Call" \t " "§):** A Switching Function communications relationshipXE "Communications relationship:See Call" \t " "§. Typically, a Call is a communications relationship between two or more Devices. Note, however, during call set-up and release, there may be only one Device on the Call. A Call is a TSAPI programming object.

**Call IdentifierXE "TSAPI:Programming handle:See Call Identifier"** \t " "§**:** A TSAPI programming handleXE "Programming handle:See TSAPI programming handle" \t " "§ that identifies a Call

**Complex CallXE "Call:Complex"§:** A Call connecting more than two Devices.

**Computing DomainXE "Computing:Domain"§:** Those computers (and their Objects) accessible from a Switching FunctionXE "Switching:Function"§. Where a switch has multiple CTI linksXE "CTI:Link"§ to multiple computers, the Computing Domain is the union of all computers connected to the switch.

**Computing FunctionXE "Computing:Function"§:** A computer, or other resource in a Computing Sub-domainXE "Computing:Sub-domain"§.

**Computing Sub-DomainXE "Computing:Sub-domain"§:** Those computers (and their Objects) accessible from the Switching FunctionXE "Switching:Function"§ using a specific CTI linkXE "CTI:Link"§. Where a switch has multiple CTI links to multiple computers, the Computing Sub-Domain is a subset of the Computing DomainXE "Computing:Domain"§. Where the switch has a single CTI link, the Computing Sub-Domain is equivalent to the Computing Domain.

**Connection**XE "Connection"§ **(TSAPI programming object**XE "TSAPI:Programming object:See Connection" \t " "§**):** A relationship between a CallXE "Call"§ and a DeviceXE "Device"§. A Connection is in one of a number of statesXE "Connection:State"§ (alerting, held, connected, etc.). Note that when a Call connects (for example) three Devices, there are three Connections for the Call. Each Connection reflects the state of the Call at one of the Devices.

**Connection Identifier**XE "Connection:Identifier"§ **(TSAPI programming handle**XE "TSAPI:Programming handle:See Connection Identifier" \t " "§**):** A TSAPI programming handle that identifies a Connection. A Call IdentifierXE "Call:Identifier"§ and a Device IdentifierXE "Device:Identifier"§ comprise a TSAPI Connection Identifier.

**Device**XE "Device"§ **(TSAPI programming object**XE "TSAPI:Programming object:See Device" \t " "§**):** An ObjectXE "Object:See TSAPI programming object" \t " "§ which abstracts the interface between a user and the communications signaling in the Switching Function. A Device can be a single endpoint (such as a telephone), or multiple endpoints that form a group (ACDXE "ACD"§ group or trunk groupXE "Trunk group"§).

**Device Identifier**XE "Device:Identifier"§ **(TSAPI programming handle)**XE "TSAPI:Programming handle:See Device Identifier" \t " "§**:** A TSAPI programming handle that identifies a Device.

**Directory NumberXE "Directory number"§:** The phone number for a Device. Directory Numbers are typically denote telephone station Devices, but ACDXE "ACD"§ groups and other DevicesXE "Device"§ may have Directory Numbers also.

**DomainXE "Domain"§:** The union of a Switching DomainXE "Switching:Domain"§ and a Computing DomainXE "Computing:Domain"§.

**EventXE "Events"§:** A stimulus of interest to an Application that (typically) causes a change in the stateXE "Device:State:State change"§ of a DeviceXE "Device"§ object.

**Event ReportXE "Event report"§:** A message from a Switching Sub-DomainXE "Switching:Sub-domain"§ to a Computing Sub-DomainXE "Computing:Sub-domain"§ indicating that an EventXE "Events"§ has occurred.

**Held CallXE "Call:Held"§:** A call (at a station) that is held (in a hold state) at that station. More specifically, the Connection (see *Connection*) for a Held Call is in the Hold State (see *TSAPI Connections* section, *Connection State* definition).

**Interconnection Service BoundaryXE "Interconnection Service Boundary"§:** An abstraction of the boundary between the Switching DomainXE "Switching:Domain"§ and the Computing DomainXE "Computing:Domain"§. In practice, CTI linksXE "CTI:Link"§ bridge the Interconnection Service Boundary.

**Object:** TSAPI programming objects include ConnectionsXE "Connection"§, CallsXE "Call"§, and DevicesXE "Device"§. Each has a corresponding programming handle, or identifierXE "Identifier:See TSAPI Programming Handle" \t " "§.

**PartyXE "Party"§:** A telephony user. A Party may be a human, applicationXE "Applications"§, or other resource (such as a port on a voice response unit).

**ServiceXE "Service"§:** The benefit provided by an ApplicationXE "Applications"§ to a User.

**Service BoundaryXE "Service:Boundary"§:** A specific CTI interfaceXE "CTI:Link"§ between a Computing FunctionXE "Computing:Function"§ and a Switching FunctionXE "Switching:Function"§. All Service boundaries cross the Interconnection Service Boundary.

**StateXE "State:**See Connection **State"** \t " "**§XE "State:See Device State"** \t " "**§XE "State:See Call State"** \t " "**§:** An object's current condition. Specifically, TSAPI ConnectionsXE "Connection:State"§ have an associated state.

**Switching DomainXE "Switching:Domain"§:** Those switches (and their Objects) accessible from a Computing FunctionXE "Computing:Function"§. Where a computer has multiple CTI linksXE "CTI:Link"§ to multiple switches, the Switching Domain is the union of all switches connected to the computer.

**Switching FunctionXE "Switching:Function"§:** A switch in a Switching Sub-domainXE "Switching:Sub-domain"§.

**Switching Sub-DomainXE "Switching:Sub-domain"§:** Those switches (and their Objects) accessible from a Computing Function XE "Computing:Function "§using a specific CTI linkXE "CTI:Link"§. Where a computer has multiple CTI links to multiple switches, the Switching Sub-Domain is a subset of the Switching DomainXE "Switching:Domain"§. Where the computer has a single CTI link, the Switching Sub-Domain is equivalent to the Switching Domain.

**UserXE "User"§:** A person, process or piece of equipment that receives direct benefit (e.g. new feature, improved performance) from an Application'sXE "Applications"§ Services.

### Acronyms

**ACDXE "ACD"§**: Automatic Call Distribution

**CSTAXE "CSTA"§**: Computer-Supported Telecommunications Applications

**ID**: Identifier

**ISDNXE "ISDN"§**: Integrated Services Digital Network

## ArchitectureXE "Architecture"§

This section summarizes the functional architecture underlying ECMA CSTAXE "CSTA"§ and TSAPI. CSTA defines the interworking between Computing and Switching FunctionsXE "Computing:Function"§XE "Switching:Function"§ in a way which is independentXE "Switch:Independent"§ of their physical implementation. This section introduces the concepts of:

- distributionXE "Distribution"§ of Computing and Switching Functions
- TSAPI Service
- client/server modelXE "Client/server model"§
- and TSAPI objectsXE "TSAPI:Programming object"§

An ApplicationXE "Applications"§ is a co-operative process distributed between a Switching FunctionXE "Switching:Function"§ (switch) and a Computing FunctionXE "Computing:Function"§ (computer). This section describes the interactions between them.

## Distribution of ComputingXE "Distribution"§ and Switching Functions

One (or several computers in a computing network) provide the Computing FunctionsXE "Computing:Function"§ and one (or several) switches provide the Switching FunctionsXE "Switching:Function"§ for a TSAPI ApplicationXE "Applications"§. The TSAPI application appears to a User (human or machine) to be a single application, not as two separate functions on two separate networks (as it is, in fact, implemented).

Since the applications use distributed resources, communications must occur between the distributed entities. Figure 3-1 shows an abstract communications model. Note that each of the distributed functions is expanded into:

- a clientXE "Client"§ application component that provides the TSAPI interactions
- a serverXE "Server"§ communications component that exchanges messages

- networking support, or lower layer interconnection

**Figure 3-2**
Model Showing the Relationship Between TSAPI Elementstc "Model Showing the Relationship Between TSAPI Elements" \f f \l3§
µ §

Figure 3-3 shows that distributed Application components use TSAPI definitions to interact with their peers. TSAPI defines the Service descriptionsXE "Service:Description"§ and provides the service interface between the Application functions and the Server providing communications with the switch. TSAPI supports various switches, and as a consequence, some of the TSAPI elements are optional and their use is implementation dependentXE "Switch:Specific"§.

## API ServicesXE "Service"§

Unless otherwise qualified, the TSAPI definition uses the term 'Service' to refer to the benefit that an application serverXE "Server"§ provides to a clientXE "Client"§ applicationXE "Applications"§. TSAPI Services are independent of the specific CTI linkXE "CTI:Link"§ connecting the switch with the application server. Since TSAPI is independent of the particular telephone terminal types, the Switching FunctionXE "Switching:Function"§XE "Switch:Specific"§ must determine how to support a given TSAPI request for its specific telephone types. For example, TSAPI does not specify how to provide the Make Call Service for analog or ISDN telephones. A Switching Function will use its existing service definitions to provide TSAPI Services on telephones where that service already exists.

TSAPI definitions do not embody the specific details of how the Switching FunctionXE "Switching:Function"§ accomplishes TSAPI Services. The Switching Function does provide an abstraction to the TSAPI Service requester, via Event ReportsXE "Event report"§, of the steps taken to accomplish the Service.

## ServicesXE "Service"§ and ObjectsXE "TSAPI:Programming object"§

A serverXE "Server"§ provides services to a clientXE "Client"§ that consist of monitoring and controlling Switching Sub-Domain XE "Switching:Sub-domain"§objects. TSAPI defines the client applicationXE "Applications"§ interface for monitoring and controlling these objects.

## Functions

A Domain XE "Domain"§is the union of a Switching DomainXE "Switching:Domain"§ and a Computing DomainXE "Computing:Domain"§. In other words, it is those switching and computing objectsXE "TSAPI:Programming object"§ that an applicationXE "Applications"§ can access. Figure 3-4 shows an example of a Domain. A heavy line divides the Domain into Switching and Computing Domains. The Switching Domain consists of Switching FunctionsXE "Switching:Function"§ S1, S2, and S3. Similarly, the Computing Domain consists of Computing FunctionsXE "Computing:Function"§ C1, C2, and C3. Each Function has a view of the Domain where it resides. Note that if multiple Functions provide an application with the same view, then the Functions are in the same Sub-Domain. TSAPI Applications (shown in Figure 3-5 as "Application Domains") are distributed across at least one Switching Sub-Domain and at least one Computing Sub-Domain.

**Figure 3-6**

Domains and Sub-Domainstc "Domains and Sub-Domains" \f f \l3§tc "Domains and Sub-Domains" \f f \l3§

Computing Domain Switching Domain

### TSAPI Switching Sub-Domain Model

XE "Switching:Sub-domain:Model"§The Switching Sub-Domain Model defines an abstract view of a Switching FunctionXE "Switching:Function"§. TSAPI defines several Switching Sub-Domain Model Objects for use in Application programming, CallXE "Call"§, and ConnectionXE "Connection"§.

## TSAPI DeviceXE "Device"§

A TSAPI applicationXE "Applications"§ can monitor and control Devices of various types (including telephones). However, a TSAPI application may not be able to monitor or control all Devices. In CSTA, a Device can refer to either a physical device (such as buttons, lines, trunks, and stations) or a logical deviceXE "Logical:Device"§ (such as groups of devices, pilot numbers, and ACDs). Devices have associated attributesXE "Device:Attribute"§, which allow applications to monitor and control them.

TSAPI Device attributesXE "Device:Attribute"§ are:

1. **Device Type**XE "Device:Type"§ - A Device has one of the following types:

   ◆ ***ACD***XE "Device:Type:ACD"§ - An Automatic Call Distributor (ACD) is a Switching FunctionXE

"Switching:Function"§ mechanism that distributes calls to ACD agents. An ACD (as opposed to ACD-group) consists only of the distribution mechanism and not the ACD agents (or their Devices) to which the mechanism can distribute calls.

- ***ACD group***XE "Device:Type:ACD group"§ - An Automatic Call Distributor (ACD) group is the mechanism that distributes calls within a Switching Function as well as the ACD agent Devices to which that mechanism distributes calls.
- ***Button***XE "Device:Type:Button"§ - is an instance of a call manipulation point at an individual station. Simple analog stations often have no physical buttons but behave as if they had one. Some advanced stations can emulate several analog stations and often represent those stations with several buttons. In some situations it is desirable to identify a given button on a multi-button station. Note that a station with several line appearance buttons could have either the same telephone number or different telephone numbers assigned to those buttons.
- ***Button group***XE "Device:Type:Button group"§ - is two or more instances of a Button at an individual station.
- ***Line***XE "Device:Type:Line"§ - is a communications interface to one or more stations typically associated with a directory number. In some situations it may be impossible to identify individual stations that share a line (a single directory number).
- ***Line group***XE "Device:Type:Line group"§ - is a set of communications interfaces to one or more stations.
- ***Operator*** XE "Device:Type:Operator "§- also known as AttendantXE "Attendant"§, is a device that is used to interact with a party to assist in call setup or to provide other telecommunications service. This device is different from other devices in that it is often

involved in setting up other calls, and is usually not part of the call after the call is connected.

- ***Operator group***XE "Device:Type:Operator group"§ - two or more operator devices used interchangeably or addressed identically.
- ***Station***XE "Device:Type:Station"§ - is the traditional telephone device. A station is a physical unit of one or more buttons and one or more lines.
- ***Station group***XE "Device:Type:Station group"§ - is two or more stations used interchangeably or addressed identically.
- ***Trunk***XE "Device:Type:Trunk"§ - a device that spans switching sub-domainsXE "Switching:Sub-domain"§. In order to monitor and control calls that cross switching sub-domains it may be desirable to address the point at which the call crosses the boundary. This point is generally a trunk or trunk group.
- ***Trunk group***XE "Device:Type:Trunk group"§ - often, many trunks connect to the same place. These trunks are often placed in groups and accessed using a single identifier. In such a configuration the individual trunks are used interchangeably.

2. **Device Class**XE "Device:Class"§ - An application may monitor or control TSAPI Devices in the various Device Classes in different ways. A Device must belong to one, and may belong to more than one, of the following classes:

- ***Data***XE "Device:Class:Data"§ - a device that is used to make digital data calls (either circuit switched or packet switched). This class includes computer interfaces and G4 facsimile machines.
- ***Image*** XE "Device:Class:Image "§- a device that is used to make digital data calls involving imaging, or high speed circuit switched data in general. This class includes video telephones and CODECs.

- ***Voice***XE "Device:Class:Voice"§ - a device that is used to make audio calls. This class includes all normal telephones, as well as computer modems and G3 facsimile machines.
- ***Other*** XE "Device:Class:Other "§- a type of device not covered by data, image, or voice.

3. **Device Identifier****XE "Device:Identifier"§** - a TSAPI programming handle for a Device that allows an applicationXE "Applications"§ to uniquely identify each device at the API. Devices are identified using staticXE "Device**:Identifier**:Static"§ and/or dynamicXE "Device:Identifier:Dynamic"§ identifiers:

   - ***Static Device Identifier*** XE "Device:Identifier:Static"§- A Static Device Identifier is stable over time. It remains constant and unique over calls. A Static Device Identifier is typically the dialed number for the Device known by both the ComputingXE "Computing:Function"§ and Switching FunctionsXE "Switching:Function"§.

     It is sometimes useful for the Switching Function to convert long phone number identifiers to another, usually shorter, static form for subsequent use in service interaction. An example of this would be the transformation of a Public Directory Number to a Private Directory Number.

     This transformation allows service interactions to be independent of the identification mechanism and allows reduction in the amount of data exchanged. This transformed number is known as a Short Form Static Device IdentifierXE "Device:Identifier:Static Short Form"§.

     Some Switching Functions allow the same dialed number to be assigned to Devices of different types. Thus, a TSAPI application may also need to use the Device Type to uniquelyXE "Device:Identifier:Uniqueness"§ address the Device.

- ***Dynamic Device Identifier*** XE "Device:Identifier:Dynamic"§- A Switching FunctionXE "Switching:Function"§ may not always make a Static Device Identifier available for every Device on a call. This may occur because a static identifier may not be available (there is no dialed number identifier for the device), or because a dialed number does not unambiguously refer to a single device (i.e. a group identifier). In these cases the Switching Function assigns a Dynamic Device Identifier as a handle for the Device for the duration of the call. Management of the Dynamic Device Identifier is discussed in a later section, *Dynamic Identifier Management*.

4. **Device StateXE "Device:State"§** - is a list of the Connection StatesXE "Connection:State"§ for all the calls which are associated with the Device. For information about Connection states see *TSAPI Connections* later in this chapter.

## CallXE "Call"§

TSAPI applicationsXE "Applications"§ can monitor and control calls (including call establishment and release). In certain operations, such as conference and transfer, one Device in a Call is replaced with another Device or two Calls merged into a single Call. In these situations, the TSAPI Call object is maintained as long as the communications relationship remains across each operation (i.e. the call survives transfer, conference, and forwarding operations). TSAPI Call object attributes are:

1. **Call Identifier**XE "Call:Identifier"§ - a Call Identifier is a TSAPI programming handle that the Switching FunctionXE "Switching:Function"§ assigns to each Call. The Call ID may or may not be unique among all calls within a Switching Sub-Domain, but coupled with a Device IDXE "Device:Identifier"§ the pair will form a uniqueXE "Call:Identifier:Uniqueness"§ Connection IDXE "Connection:Identifier"§ within a Switching Sub-DomainXE "Switching:Sub-domain"§. To allow reference to a nascent call, The switch will assign a Call ID before a call is fully established. For example, a switch will assign a Call ID to an incoming call when the called Device is Alerting (the assignment is done before the call is answered).

   Certain Services merging multiple calls into a single call.. Examples of such TSAPI Services are Transfer and Conference.  During operations of Services that merge multiple calls, the call identifier may change, but the call continues as a TSAPI object. The management of the call identifier is described in a later section, *Dynamic Identifier Management*.

2. **Call state** XE "Call:State "§- is a list of the Connection statesXE "Connection:State"§ for all the DevicesXE "Device"§ that are a part of the CallXE "Call"§.

   For simplicity, common call statesXE "Call:State:Common "§ for two-party calls have been a single descriptive name. For example, a two party call with a Connection State of "Connected" at one station and a Connection State of "Alerting" at the other has a Call State of "Delivered". Table 3-1 gives the mapping of descriptive names to Connection State lists for two-party calls. station are provided as a single value, whereas uncommon call states are provided as a list. For more information on Connection

States, see the following section, *TSAPI Connections*. The *Call States* section of this chapter gives further information about Call States.

## TSAPI ConnectionsXE "Connection"§

A Connection is a relationship between a CallXE "Call"§ and a DeviceXE "Device"§. A TSAPI applicationXE "Applications"§ can monitor or control a Connection. For example TSAPI Services Hold Call, Reconnect Call, and Clear Call all control Connections. Connections are TSAPI programming objects with the following attributesXE "Connection:Attributes" \t " "§:

1. **Connection Identifier**XE "Connection:Identifier"§ – is a TSAPI handle that is made up of a Call IdentifierXE "Call:Identifier"§ and Device IdentifierXE "Device:Identifier"§. For a call there are as many Connection identifiers as there are associated devices. Similarly, for a device there are as many Connection identifiers as there are associated calls. The Connection IdentifierXE "Connection:Identifier:Uniqueness"§ is unique within a Switching Sub-DomainXE "Switching:Sub-domain"§ and within a single TSAPI server. A TSAPI application cannot use a Connection Identifier until it has received the identifier from the Switching Function.

2. **Connection State**XE "Connection:State"§ - is the state of a call at a Device. The Connection state always refers to a single Call/Device relationship. Snapshot Services report Connection States for Calls and Devices. Monitors report Events, which are changes in Connection States for the monitored entity.  Figure 3-7 shows a sample Connection state model. Note that since TSAPI is switch independentXE "Switch:Independent"§, and since switch features vary from switch to switchXE "Switch:Specific"§ (and therefore interact differently on different switches),

there is no definitive TSAPI Connection State model to which all switches comply.

**Figure 3-8**

Sample Connection State Modeltc "Simple Connecttion State Model" \f f \l3§tc "Sample Connection State Model" \f f \l3§

The transitions between states, shown by arrows, form the basis for providing Event Reports.

The TSAPI Connection statesXE "Connection:State"§ are defined as follows.

- ***Null***XE "Connection:State:Null"§ - the state where there is no relationship between the call and device.
- ***Initiated***XE "Connection:State:Initiated"§ - the state where the device is requesting service. Usually this results in the creation of a call. Often this is thought of as the "dialing" state.
- ***Alerting***XE "Connection:State:Alerting"§ - the state where a device is alerting (ringing).
- ***Connected*** XE "Connection:State:Connected "§- the state where a device can communicate with other Devices on a call (cannot be a held call).
- ***Held*** XE "Connection:State:Held"§- the call is "on hold" at the Device.
- ***Queued*** XE "Connection:State:Queued "§- the state where normal state progression has been stalled. For example, a

call being processed by an ACD that is waiting for an ACD agent to become available is "queued".

- ***Failed*** XE "Connection:State:Failed "§- the state where normal state progression has been aborted. a "Failed" state can result because of failure to connect to the calling (originator) device, failure to connect the called (destination) device, failure to create the call, and other reasons.

## Call Status Event ReportsXE "Call status event reports:See Event reports" \t " "§XE "Event report"§

The Switching Sub-DomainXE "Switching:Sub-domain"§ model is an abstract view of call states and events. This abstract view is probably more detailed than most applicationsXE "Applications"§ require, but it introduces a precise language for describing Event Reports, Call States, and Service functional descriptions. Connection stateXE "Connection:State"§ changes correspond to telecommunications signaling at a DeviceXE "Device"§.

ISDNXE "ISDN"§ specifications model network access as a distributed state machine. ECMA CSTAXE "CSTA"§ borrows from this ISDN model. One part of this access state machine resides in the DeviceXE "Device"§. There is another similar distributed access state machine which resides across the ISDN network at the egress device.

Thus, a call can be modeled as a collection of Connection stateXE "Connection:State"§ machines. Network signaling causes changes in the state machines across the network. When signaling occurs, a state change occurs at the affected ConnectionXE "Connection"§. Figure 3-9 shows this concept of communication between distributed state machines for the case of establishing a simple call. For informational purposes, the ISDNXE "ISDN"§ call states have also been shown.

**Figure 3-10**

Relationship of Call State Changes and Event Reports to Network Signalingtc "Relationship of Call State Changes and Event Reports to Network Signaling" \f f \l3§

| Time |  | Device D1 | Call C1 | Device D2 |  |
|---|---|---|---|---|---|
| T1 | Null | Null | No Event Report | Null | Null |
| T2 | Setup | Initiated | Service Initiated | Null | Null |
| T3 | Proceeding | Connected | Originated | Null | Null |
| T4 | Delivered | Connected | Delivered | Alerting | Receive |
| Te | Connected | Connected | Established | Connected | Connected |

Notice in Figure 3-11 that the StateXE "Connection:State"§ changes and Event ReportsXE "Event report"§ are based on Switching FunctionXE "Switching:Function"§ signaling. Many simple Connection events are of interest to applicationsXE "Applications"§. Certain telecommunications operations involve changes to many Connections. TSAPI reports these compound events (such as Transfer, Conference and Clear Call) in a single Event Report. Each TSAPI Event Report defines which Connections have changed state.

## TSAPI Call StatesXE "TSAPI:Call States:See Call State" \t " "§XE "Call:State"§

A Call State is defined as the list of Connection statesXE "Connection:State"§ for all the DevicesXE "Device"§ involved in the call. This list is also called the **Compound Call StateXE "Call:State:Compound"§**. Listing the Connection states can describe any possible call state. However, most calls are often in a small number of widely recognized states.

TSAPI defines those states as the **Simple Call States** XE "Call:State:Simple"§shown in Table 3-1. TSAPI does not report Simple Calls States as a list, but rather in an abbreviated fashion. Note that Simple Call States can differ by the order of the Connection state list. Alerting-Connected is not equal to

Connected-Alerting. The first is the Simple Call State "Received" and the second is "Delivered".

Null can be a known Connection stateXE "Connection:State"§, so for a nascent call it is possible to have a Call state with only one non-Null Connection (see Table 3-1).

For calls with two Connections, the Table 3-2 summarizes the Simple Call StatesXE "Call:State:Simple"§ assigned to the combinations of Connection States. If there is no entry in Table 3-3 for the combination of Connection states, then TSAPI provides the list.

For calls with more than two non-Null Connection states, The Call State is a compound call state**XE "Call**:**State:Compound"§**. TSAPI (at times) simplifies the compound call state by relating it to a particular device. The Connection State related to a particular device in this way is called the **Local Connection StateXE "Connection**:**State:Local"§**. Other Connection States are not differentiated from one another. A three party conference call that is on hold at a given Device and connected to the other two devices has a Local Connection State of "Held" at that given Device.

**Table 3-4**

TSAPI Simple Call StatesXE "Call:State:Simple"§tc "Simple Call States" \f t \l3§

| Local Connection State | Other Connection State | Simple Call state |
|---|---|---|
| Alerting | Connected | Received |
| Alerting | Hold | Received-On Hold |
| Connected | Alerting | Delivered |
| Connected | Connected | Established |
| Connected | Failed | Failed |
| Connected | Hold | Established-On Hold |
| Connected | Null | Originated |
| Connected | Queued | Queued |

| | | |
|---|---|---|
| Hold | Alerting | Delivered-Held |
| Hold | Connected | Established-Held |
| Hold | Failed | Failed-Held |
| Hold | Queued | Queued-Held |
| Initiated | Null | Pending |
| Null | Null | Null |

## Dynamic Identifier ManagementXE "Device:Identifier:Dynamic"§

Since Connection IdentifiersXE "Connection:Identifier"§ comprise a Device IDXE "Device:Identifier"§ and a Call IDXE "Call:Identifier"§, proper management of Connection Identifiers will, in turn, provide proper management of Dynamic Device IdentifiersXE "Device:Identifier:Management"§ and Call IdentifiersXE "Call:Identifier:Management"§.

The Switching FunctionXE "Switching:Function"§ provides Connection IdentifiersXE "Connection:Identifier"§ when either a new CallXE "Call:Identifier"§ or Device IdentifierXE "Device:Identifier"§ is needed. When a call is made the switch provides a Connection Identifier. The switch then provides the Connection ID in any following Event ReportsXE "Connection:Identifier:In event report"§ that pertain to that call. Similarly, the switch provides Connection IDs containing a Device ID for a device involved in a call.

The switch updatesXE "Connection:Identifier:Update"§XE "Connection:Identifier:Update"§XE "Device:Identifier:Update"§ identifiers when needed. If a Conference or Transfer (merging two calls) changes a Call ID, then the switch provides Event Reports containing Connection IDs that link the old call identifier to the new identifier. Similarly, if a Dynamic Device Identifier is changed, the switch -provides new Connection Identifiers for the devices in the call. Both Service Acknowledgments and Event Reports may contain information necessary to manage identifiers.

Identifiers cease to be validXE "Connection:Identifier:Invalid"§XE "Connection:Identifier:Invalid"§XE "Device:Identifier:Invalid"§ when their context vanishes. If a call ends, its call identifier is no longer valid. Similarly, if a device is removed from service or from a call, its dynamic device identifier becomes invalid. Many Event Reports and Services specify when a Connection Identifier has lost or will lose its context.

Identifiers can be reusedXE "Connection:Identifier:Reuse"§XE "Connection:Identifier:Reuse"§XE "Device:Identifier:Reuse"§. Once an identifier has lost its context it may be re-used to identify another object. Most implementations will not reuse identifiers immediately.

Call and Device Identifiers can be, but are not guaranteed to be, globally uniqueXE "Connection:Identifier:Uniqueness"§XE "Connection:Identifier:Uniqueness"§XE "Device:Identifier:Uniqueness"§. The TSAPI server ensures that the combination of Call and Device Identifier is globally unique within a Switching Sub-DomainXE "Switching:Sub-domain"§. To accomplish this, compliant PBX drivers ensure that either the call identifier, or the device identifier (or both) is globally unique. In many cases the Connection IdentifierXE "Connection:Identifier"§ requires the use of both the CallXE "Call:Identifier"§ and Device IdentifiersXE "Device:Identifier"§ to uniquely refer to Connections in a call.