



TELEPHONY SERVICES

August 2000





TELEPHONY SERVICES

A	REFERENCE	5
1	Telephony Services on The BinTec router	6
1.1	Overview	6
1.2	What is POTS?	7
1.2.1	POTS Interfaces	8
1.3	What is TAPI?	12
1.3.1	Remote TAPI on the BinTec router	13
1.4	Configuring Telephony Services	15
1.4.1	Two workspaces: two telephones, one VICAS	15
1.4.2	One workspace: one VICAS, one telephone, one fax	19

REFERENCE

1 Telephony Services on The BinTec router

1.1 Overview

Telephony Service on the BinTec router means that you can connect conventional analog devices (telephone, fax, modem, etc.) to the BinTec router and place or receive analog calls via any of the BinTec router's ISDN interfaces.

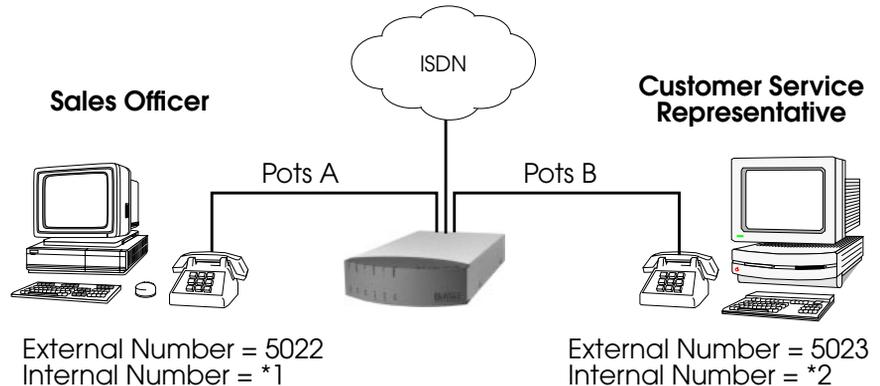
NOTE:

System software support for telephony services is included on all BinTec router products. However, to take advantage of telephony services, a CM-AB module must be installed.

This allows you to use the BinTec router as a [PBX \(Private Branch exchange\)](#) to:

1. Make toll-free calls internally between connected analog devices, or
2. Place (or receive) calls via the ISDN from connected analog devices.

This could be useful for small offices when combined with PCs running Remote TAPI Client software (included on the Companion CD).



In the simple scenario shown above, the BinTec router is used to make inter-office (toll-free) calls between workstations using the internal telephone numbers. In addition, both parties can simultaneously place or receive calls from the ISDN using their respective analog devices.

Although telephony services and the Remote TAPI go together hand in hand, the rest of this chapter describes using the BinTec router's [POTS ports](#) and using the BinTec router as a Remote [TAPI Server](#) separately.

1.2 What is POTS?

In the networking field the term [POTS \(Plain Old Telephone Service\)](#) is often used to refer to the conventional analog telephone network or analog-based communications devices. With the CM-AB module installed the BinTec router has two POTS ports on its backplane called POTS A and POTS B, for the attachment of such devices (analog telephone, FAX machine, or modem).

1.2.1 POTS Interfaces

When the CM-AB module is installed an entry in the *bi-boAdmBoardTable* will display the slot number where the board is installed. On the VICAS and the BRICK-XS Office slot 3 is always used. Information about the devices connected to the POTS ports is stored in the *potsIfTable*. On the VICAS the table looks as follows.

```
mybrick: > potsIfTable
```

inx Slot(*ro)	Unit(*ro)	Type(rw)
00 3	0	any
01 3	1	any

```
mybrick : potsIfTable>
```

The fields of the *potsIfTable* have the following meanings.

- Slot** Identifies the slot the CM-AB module is installed in.
- Unit** Identifies the port, POTS A = Unit 0, POTS B = Unit 1.
- Type** Identifies the types of calls this device will accept.
Possible values include: any, telephony, fax, modem, or disable.
Disable means that the device can not place or accept calls.

These are the default entries created by the system at boot time upon detection of an installed CM-AB module. *potsIfTable* entries can only be removed by the system.

Dispatching Analog Calls

The BinTec router dispatches incoming calls (from the ISDN) according to the [ISDN Call Dispatching](#) algorithm. The dispatching algorithm distributes calls to BinTec router services according to the "Called Party's Address" contained in the call packet and the localNumber field of the *isdnDispatchTable*. Calls dispatched to the pots service (*Item=pots*) are given to POTS devices based on additional information contained in the service indicator field of the ISDN Call packet and the *Type* field mentioned above. The service indicator field simply specifies the type (FAX, voice, data , etc.) of call.

NOTE



If the call originated from an analog device, the ISDN can't always accurately report the call type and simply reports the call as being an "analog" call; the actual call may be a FAX or voice call.

The different call types and the services they support are as follows:

<i>potsType</i>	Accepts calls from	
	analog network	ISDN devices
any	analog	telephony, fax, modem
telephony	analog	telephony
fax	analog	fax
modem	analog	modem

Internal Calls

Internal calls can be made between devices connected to the BinTec router's POTS ports. This requires that each POTS device in the connection is assigned an internal number). Note that these calls are dispatched according to the *isd-*

nDispatchTable; therefore it's recommended that you assign internal numbers using the format "**<internal number>*" to ensure internal and external [MSNs](#) are kept separate. Internal numbers are assigned to devices in the *isdnDispatchTable* as follows.

<i>isdnDispatchTable</i>	POTS A	POTS B
<i>StkNumber</i>	31	
<i>Item</i>	pots	
<i>LocalNumber</i>	<the telephone number for this device>	
<i>Bearer</i>	any	
<i>Slot</i>	<slot number for CM-AB module>	
<i>Unit</i>	0	1
<i>Direction</i>	both	
<i>Mode</i>		
<i>UserName</i>		

NOTE

By default the BinTec router automatically creates two dispatch table entries upon detection of the CM-AB module at boot time. The default internal numbers for ports A and B are "*1" and "*2" respectively.

External Calls

External calls can also be placed or received from POTS devices. External numbers are assigned to POTS devices in the same way as for internal numbers with the exception that the *StkNumber* field must specify a "real" ISDN Stack number.

Outbound External Calls

Each POTS device must have exactly one outgoing number for outbound external calls. This means either

1. an external number entry with the *Direction* field set to **both** OR
2. an external number entry with the *Direction* set to **outgoing**.

Inbound External Calls

A POTS device may be configured to respond to different external numbers by creating multiple external-number entries in the dispatch table and setting the *Direction* field to **incoming**.

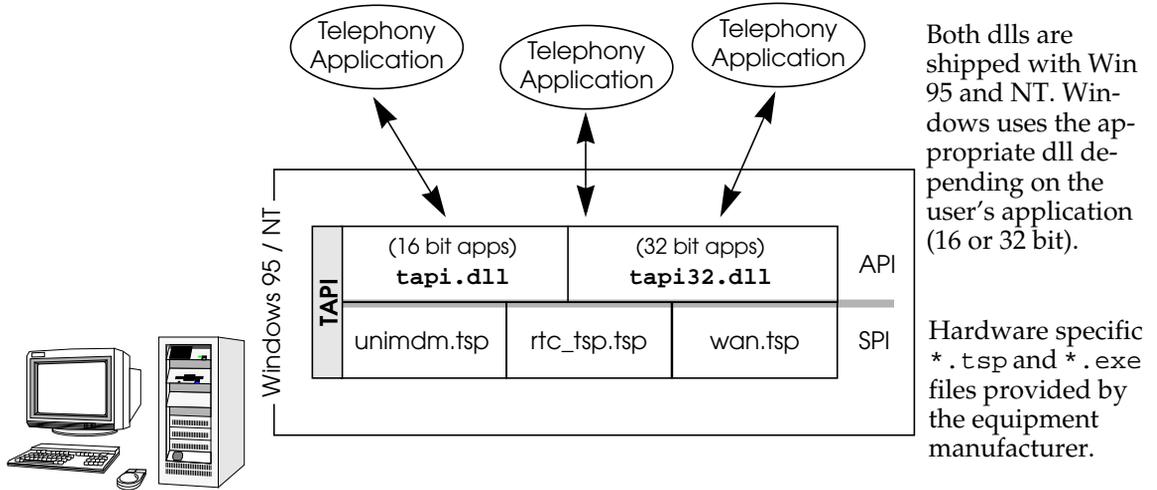
1.3 What is TAPI?

[TAPI \(Telephony Applications Programming Interface\)](#) is a programming interface initially defined by Microsoft and Intel for developing Windows-based telephony applications. A telephony application uses attached telephone equipment to place, accept, or monitor calls. The Microsoft Dialer (part of Windows) is an example of a TAPI application.

TAPI actually consists of two parts.

1. The API defines how applications (like the Microsoft Dialer) interact with the underlying operating system (Windows 95 or NT). It gives applications access to Windows' telephony features.
2. The SPI (Service Provider Interface) defines how the operating system interacts with attached telephony hardware. More than one [TSP \(Telephone Service Provider\)](#) may be installed on the PC, each one specifies how the OS communicates with a particular piece of hardware.

TAPI on Windows 95 and NT systems looks like this.

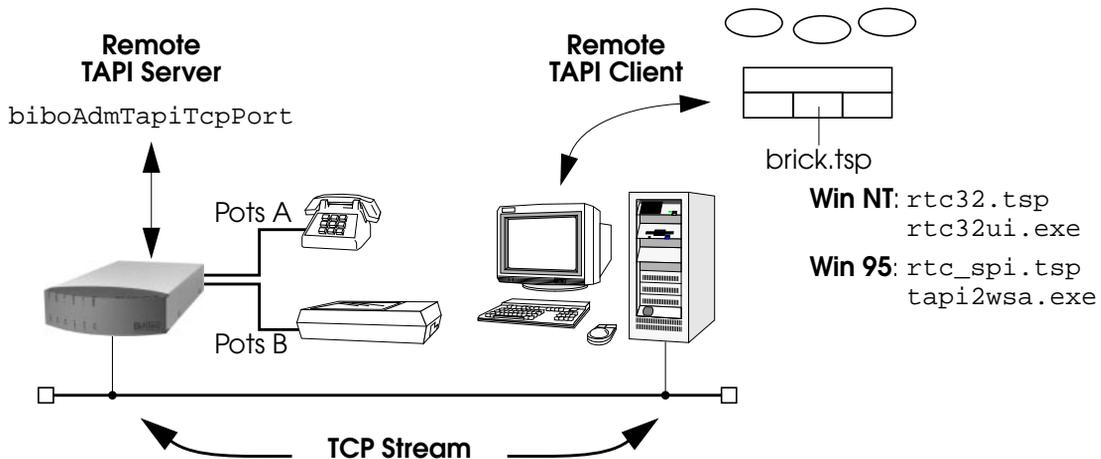


1.3.1 Remote TAPI on the BinTec router

The BinTec router can be used as a remote TAPI server, meaning that it can place, accept, and route calls from a PC on the LAN where the Remote TAPI Client is installed. Remote TAPI Client software for PCs is included on the Companion CD and is installed from the BRICKware installation program.

The Remote TAPI Client forwards all TAPI requests made by TAPI applications to the BinTec router via a TCP stream.

The BinTec router accepts TAPI client connections via it's TAPI port.



TAPI Settings

Configuring the BinTec router as a TAPI server is straightforward. On the BinTec router, all that is required is that the BinTec router's TAPI port be set. This is defined in the *admin* table.

The *biboAdmTapiTcpPort* variable defines the TCP port on the BinTec router remote TAPI applications must connect to. By default the BinTec router uses TCP Port 6001. The same value must be configured on the PC running the Remote TAPI Client program.

The TAPI server can be disabled completely by setting the TAPI port to 0.

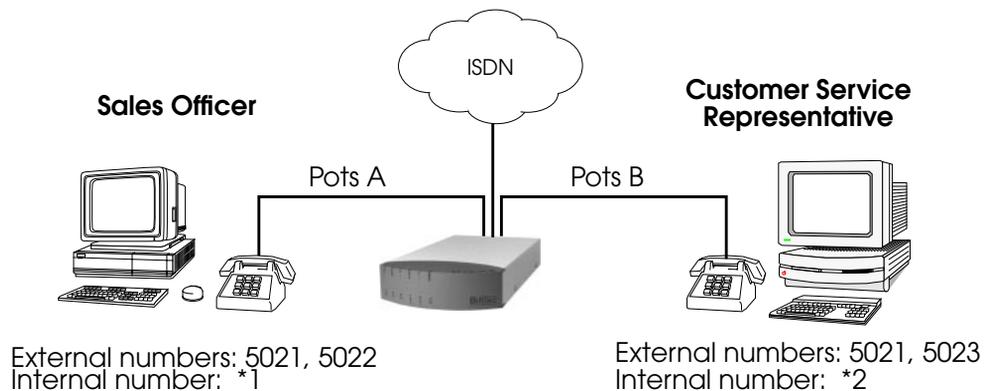
1.4 Configuring Telephony Services

Below are two example configurations showing how the BinTec router can be used as a simple PBX. The Remote TAPI Client could be installed on the computers in these examples allowing calls to be managed directly from the PC.

1.4.1 Two workspaces: two telephones, one VICAS

Here we have two workspaces, a Sales Agent and a Customer Service Representative. Each workspace has an analog telephone that is connected to the BinTec router. The sample configuration shows the entries that would be made to the *isdnDispatchTable* to allow:

1. Both workspaces receive incoming ISDN calls placed to 5021, i.e., both phones ring, the first one to pick up gets the call.
2. Each workspace has a separate number for direct calls.
3. Internal calls can be placed between workspaces.



Step 1

Create the internal numbers by locating the entries for the POTS A and POTS B interfaces in the *isdnDispatchTable*. There will be two entries that use *Stack 31*. You can identify the A and B interfaces by the *Slot:Unit* combination. POTS A is always at Unit 0, and POTS B at Unit 1. By default, POTS A uses the internal number *"*1"* while POTS B uses *"*2"*. Since the default values are what we want for our our example setup, we don't need to change them.

```
mybrick: > isdnDispatchTable
inx StkNumber(*rw)      Item(*-rw)      LocalNumber(rw)
  LocalSubaddress(rw)  Bearer(rw)     Slot(rw)
  Unit(rw)             Direction(rw)  Mode(rw)
  UserName(rw)

00 31                  pots           "*"1"
   0                   any            3
  "default"           both          right_to_left

01 31                  pots           "*"2"
   1                   any            3
  "default"           both          right_to_left
mybrick : isdnDispatchTable>
```

Step 2 Create an entry for each device's direct number. Only POTS A can answer calls to 5022; only POTS B can an-

swer calls to "5023". The first entry is for POTS A, the second POTS B.

```
mybrick: isdnDispatchTable > StkNumber=0 Item=pots Slot=3 Unit=0 LocalNumber=5022
    Direction=both Mode=right_to_left UserName="default"

mybrick: isdnDispatchTable > StkNumber=0 Item=pots Slot=3 Unit=1 LocalNumber=5023
    Direction=both Mode=right_to_left UserName="default"

mybrick : isdnDispatchTable> isdnDispatchTable
inx StkNumber(*rw)      Item(*-rw)      LocalNumber(rw)
  LocalSubaddress(rw)  Bearer(rw)     Slot(rw)
  Unit(rw)             Direction(rw)  Mode(rw)
  UserName(rw)

02 0                    pots           "5022"
   0                    any            3
  "default"            both          right_to_left

03 0                    pots           "5023"
   1                    any            3
  "default"            both          right_to_left

mybrick : isdnDispatchTable>
```

Step 3

Create the entries for our common external number "5021." This will allow both device to receive calls placed to this

number. Again, our first entry is for POTS A the second is for POTS B.

```
mybrick: isdnDispatchTable > StkNumber=0 Item=pots Slot=3 Unit=0 LocalNumber=5021
Direction=incoming Mode=right_to_left User-
Name="default

mybrick: isdnDispatchTable > StkNumber=0 Item=pots Slot=3 Unit=1 LocalNumber=5021
Direction=incoming Mode=right_to_left User-
Name="default

mybrick : isdnDispatchTable> isdnDispatchTable
inx StkNumber(*rw)      Item(*-rw)      LocalNumber(rw)
  LocalSubaddress(rw)  Bearer(rw)     Slot(rw)
  Unit(rw)             Direction(rw)  Mode(rw)
  UserName(rw)

04 0                    pots           "5021"
   0                    any            3
  "default"            incoming       right_to_left

05 0                    pots           "5021"
   1                    any            3
                    incoming

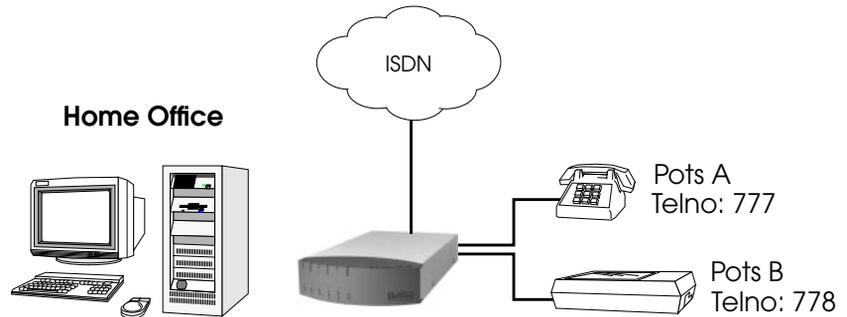
mybrick : isdnDispatchTable>
```

The configuration is complete. Don't forget to save your changes to a configuration file with **cmd=save**.

1.4.2 One workspace: one V!CAS, one telephone, one fax

In this example, we have a workspace consisting of a V!CAS, an analog telephone, and a FAX device connected to the V!CAS. In this scenario we want to configure the BinTec router so that:

1. Incoming telephony calls are given to the device at POTS A.
2. Incoming FAX calls are given to the device at POTS B.



Step 1

For this example all we need to do is create the external numbers so that both devices can be reached from the ISDN. Although we won't be making [Internal Calls](#) in this ex-

ample, the "Stack 31" entries will still be present in the *isd-
nDispatchTable*.

```
mybrick: > isdnDispatchTable
inx StkNumber(*rw)      Item(*-rw)      LocalNumber(rw)
  LocalSubaddress(rw)   Bearer(rw)      Slot(rw)
  Unit(rw)              Direction(rw)

02 0                    pots            "777"
   0                    telephony      3
   0                    both

03 0                    pots            "778"
   1                    fax             3
   1                    both

mybrick : isdnDispatchTable>
```