

# RELEASE NOTE BINGO! PLUS/PROFESSIONAL

## FEBRUARY 22, 1999

---

### New System Software:

#### *Release 4.9 Revision 4*

This document describes the new features, enhancements, bug-fixes, and changes to the BinGO! Plus/Professional System Software for Release 4.9 Revision 4.

We recommend having a look at BinTec's WWW server at <http://www.bintec.de> (Section: FTP Server), where you can always find current, up-to-the-minute information about products and software releases.

Together with this Software Release 4.9.4, a new BRICKware is available, which can be found on the current BinTec ISDN Companion CD or retrieved from BinTec's WWW server (Section: FTP Server). The features Call Pickup (page 4) and Priority Voice (page 5) are contained in this Software Release.

New System Software:.....	1	
Upgrading System Software .....	3	
What's New in Release 4.9.4 .....	4	
Features .....	4	
New BRICKware available .....	4	
PABX: Call Pickup .....	4	
Priority Voice .....	5	
Credits Based Accounting System .....	6	
Bridging .....	7	
Fax Storage: LED .....	8	
Fax Storage: msgForwardTable .....	8	
Fax Storage: Fax ID .....	9	
New Timer in x25LinkPresetTable .....	9	
Changes .....	10	
CAPI Syslog Messages .....	10	
CAPI DATA_B3_IND message .....	10	
Fax Storage: Forwarding .....	10	
Fax Storage: Forwarding Order .....	11	
PABX: Idle Tone .....	11	
PABX: Configuration during an Established Call .....	11	
Bugfixes .....	12	
Fax Storage .....	12	
Reboot when Establishing ISDN Connections .....	12	
X.25 Routing Priorities .....	13	
NAT on a Dial-Up Interface .....	13	
Setup Tool: WAN Partner Configuration .....	13	
Dynamic IP Address Pools .....	14	
RADIUS OSPF Interfaces .....	14	
Fax: T30 Carrier detection (V.21) .....	14	
Accepting Calls with CAPI 1.1 Applications .....	14	
biboAdmCapiTcpPort/biboAdmTapiTcpPort .....	15	
IPX: ripCircTable and sapCircTable .....	15	
IPX: Configuring the NetNumber .....	15	
HTTP Server: Internet Explorer 4.0 .....	15	
ISDN S <sub>0</sub> : Auto Configuration .....	16	
Bridging .....	16	
Fax Applications with Protocol Switching .....	17	
CAPI and Incorrect Bearer Capability .....	17	
TAPI: Calls Sent to TAPI Clients Several Times .....	17	
PPP Callback Working on the 2nd Attempt .....	17	
Multiple Fax Forwarding with Different Extensions ..	18	
Problems Accessing Compuserve for the First Time	18	
Known Issues .....	19	
Faulty Reorganisation in Fax Memory .....	19	
Connection Attempts between Client and Server .....	19	
Deletion of Faxes from Fax Memory .....	19	
Outgoing FTP Connections via NAT .....	19	
Autologout Interrupting the Update .....	20	
Several Extensions for one POTS Port .....	20	
Detailed Feature Descriptions .....	21	
Credits Based Accounting System .....	21	
What Was New in Release 4.9.2 .....	27	

## Upgrading System Software

1. Retrieve the current system software image from BinTec's WWW server at <http://www.bintec.de> (Section: FTP Server).
2. With this image you can upgrade the BinGO! Plus/Professional with the **update** command from the SNMP shell via a remote host (i.e. using telnet, minipad, or isdnlogin) or by using the **BOOTmonitor**, if you are logged in directly on the console.

Information on using the BOOTmonitor can be found in the *BinGO! Plus/Professional User's Guide* under *Firmware Upgrades*.

3. Please note that since software release 4.8.7, there is a new update procedure in case there is not enough memory available to perform a software update via the **update** command from the SNMP shell.

The new incremental update loads the new software image in blocks of 64 KB via TFTP and writes it to the flash ROM immediately. Because this procedure offers no possibility to check the integrity of the image, please first use the option “-v” that verifies the image file.

4. Once you've installed Release 4.9 Revision 4, you may want to retrieve the latest documentation (in Adobe's PDF format), which is also available from BinTec's file server at the address noted above.

**Note:** When upgrading system software, it is also recommended that you use the most current versions of *BRICKware for Windows* and *UNIXTools*. Both can be retrieved from BinTec's file server.

# What's New in Release 4.9.4

## Release 4.9 Revision 4:

Features:

Bugfixes:

Detailed Description:

### *Features*

#### New BRICKware available

There is a new BRICKware version 4.9.4 available. BRICKware 4.9.4 is contained on the current BinTec ISDN Companion CD or can be retrieved from BinTec's WWW server (Section: FTP-Server) at <http://www.bintec.de>

#### PABX: Call Pickup

Call Pickup was implemented as a new PABX feature into the PABX of the BinGO! Plus/Professional and V!CAS. The Call Pickup feature needs no configuration. (All extensions configured for both POTS ports are by default assigned to one group and this setting cannot be changed.)

It is possible to make use of Directed Call Pickup and Group Call Pickup for the phones that are connected to your router.

Group Call Pickup means that when there is an incoming call on a phone which is connected to one port, you can get connected to that call from a phone which is connected to second port by lifting the handset and dialing the Group Call Pickup code.

Directed Call Pickup means that the Call Pickup code which you are dialing is directed to an incoming call on a certain extension. This can be useful, for example, when a phone can be reached via two different extensions, one for private and one for business calls. When you only want to pickup the business calls, you can direct your Call Pickup to the business call extension. (Please also see [Several Extensions for one POTS Port](#) on page 20)

If a Call Pickup is not successful, you hear the occupied signal.

The following table shows the Call Pickup codes:

Code	Function
<b>*90#</b>	GROUP CALL PICKUP Group Call Pickup allows you to answer a call directed to any other extension.
<b>*90*&lt;ext.&gt;#</b>	DIRECTED CALL PICKUP Directed Call Pickup allows you to answer a call directed to a certain extension. Here dialing " <b>*90*#</b> ", i.e. leaving out the extension, has the same function as the Group Call Pickup code.

Call Pickup is also possible when you are just speaking on the one line and there is an incoming call on the telephone connected to the second port:

You place the active call on hold by pressing the **R** key (or **\*0#**), then dial the code for Directed or Group Call Pickup (**\*90#** or **\*90\*<ext.>#**) and you are connected to the incoming call. To return to the held call or to toggle between the calls, press the **R** key (or **\*0#**) again.

## Priority Voice

With this release, the Priority Voice feature is available for the products of our Voice-Data Product Line.

Priority Voice makes you reachable via the telephone connected to your router, although both B-channels are being used for data transmission to/from a WAN partner.

If both B-channels are occupied by a data connection and you want to setup an outgoing call or accept an incoming call, one B-channel of the data connection is closed down to make it available for the telephone connection.

However, you must notice that this is only possible if the two B-channels are part of the same multilink PPP connection to one WAN partner, i.e. you have configured dynamic or static channel bundling. If the two B-channels are connected to different WAN partners, the Priority Voice feature does not take effect.

Priority Voice can be configured via the SNMP shell by setting the variable ***pabxPriorityVoice*** in the table ***pabx*** to the value ***enable*** (default value: ***disable***).

The same configuration can be made via Setup Tool in the Static Settings of the PABX Menu as shown below.

BinGO! Plus Setup Tool [PABX][STATIC]: PABX Static Settings	BinTec Communications AG bingo
Dial Procedure	Prefix # for internal calls
Remote TAPI Server Port	2663
Priority Voice Feature	on
SAVE	CANCEL

The item **Priority Voice Feature** must be set to **on** to enable Priority Voice. The default value is **off**.

### Credits Based Accounting System

With dial-up WAN connections, it may occur that charges increase, because of configuration errors. The Credits Based Accounting System gives BinGO! Plus/Professional administrators the ability to control charges. It allows the BinGO! Plus/Professional administrator to limit the number of connections, the connection time and the accounted charges of every subsystem during a specified period of time. The Credits Based Accounting System can also be used to control the PABX subsystem, i.e. the POTS ports.

You can find a detailed description of this feature in “Detailed Features Description” under [Credits Based Accounting System](#).

## Bridging

Two new BinTec-specific variables are introduced to the MIB tables of the **bridge** group.

The first is the variable ***dot1dStpBridgePPPPForwardDelay*** in the table ***dot1dStp***, which allows you to time the process of the establishment of a bridging connection across the WAN.

The second variable is ***dot1dStpPortBackupForIfIndex*** in the ***dot1dStpPortTable***, which is meant to determine a dialup connection as a backup connection for an existing leased line.

Both variables are described in detail below:

### ***dot1dStpBridgePPPPForwardDelay***

The unit of the value of this variable is 1/100 seconds, the range lies between 0 and 3000 and the default value is 500 (= 5 seconds).

With ***dot1dStpBridgePPPPForwardDelay*** the user can adjust how long the port of a PPP connection is waiting until a change of the port state is allowed. This concerns the changes of state from “blocking” or “disabled” to “listening”, from “listening” to “learning” and from “learning” to “forwarding”. ***dot1dStpBridgePPPPForwardDelay*** only affects WAN connections like, for example, PPP and X.25. For LAN connections further on the value of the variable, ***dot1dStpForwardDelay*** is used to determine the changes of the port state.

When the default value is used for ***dot1dStpBridgePPPPForwardDelay*** (500), it takes 5 seconds to change the state from “listening” to “learning” and another 5 seconds to change from “learning” to “forwarding”. After an ISDN connection has been established, it consequently takes 10 seconds until data is transferred in the port state “forwarding”.

This delay is necessary for the spanning tree algorithm to detect redundant paths. If there is no redundant path to your WAN connection, the value of the variable ***dot1dStpBridgePPPPForwardDelay*** can be adjusted to “0”, i.e. the port state changes immediately from “disabled” or “blocking” to “forwarding”.

### *dot1StpPortBackupForIfIndex*

This variable is intended to be used for a setting where two BRICKs are “bridging” two LANs via PPP connections (WAN).

One connection is a leased line connection, the second a dialup connection.

To configure the dialup connection as a backup connection for the leased line connection, you must proceed as follows:

In the *dot1dStpPortTable*, set the value of the variable *dot1StpPortBackupForIfIndex* for the interface of the dialup connection to the *dot1dStpPortIfIndex* of the leased line connection. This configuration makes the dialup connection the backup connection of the leased line connection.

As long as the leased line connection is up (*dot1dStpPortState* is *forwarding*), the dialup connection is not established. When the leased line connection fails, the dialup connection is established. In the latter case, when the dialup connection substitutes the leased line connection, entries in the *dot1dTpFdbTable* belonging to the leased line connection are deleted.

### Fax Storage: LED

Concerns BinGO! Professional only.

The LED “MSG” displays a red light when a facsimile is stored in the fax storage.

### Fax Storage: msgForwardTable

Concerns BinGO! Professional only.

Two variables have been added to the *msgForwardTable*:

- *msgForwardMaxRetries*

*msgForwardMaxRetries* is the number of attempts which is made to forward faxes that are stored in the fax storage. Only attempts where a connection is established, i.e. the called partner accepts the call, are taken into account. This variable is ignored when faxes are forwarded to a local CAPI application.



If *msgForwardMaxRetries* is set to 0, there is no limitation to the number of attempts to forward faxes.

The default value for *msgForwardMaxRetries* is 5.

- ***msgForwardState***

When the number of maximum retries to forward a fax (see above) is reached, the variable *msgForwardState* is set to ***blocked***. When faxes are to be forwarded again, the user must adjust the value ***enabled*** to the variable. If no more faxes are to be forwarded, *msgForwardState* must be set to ***disabled***.

These variables can not yet be configured via the Setup Tool. If a fax can not be delivered and the maximum number of attempts is reached, the *msgForwardState* must either be reset to enabled or the fax that can't be delivered must be deleted manually (Setup Tool or SNMP shell).

## Fax Storage: Fax ID

When a fax is received by the Fax Storage, the Fax Storage communicates its own Fax ID to the sender.

The Fax ID consists of the values of the 5 MIB variables: ***pabxTrunkCountryCode***, ***pabxTrunkAreaCode***, ***pabxTrunkSubscriberNo***, ***pabxTrunkExtension*** and ***msgForwardExtension***.

## New Timer in x25LinkPresetTable

To control the state of an X.25 connection and an X.25 partner in certain time intervals, a new timer in the variable ***L2SupervTime*** has been added to the ***x25LinkPresetTable***. The value of this timer can be an integer between 100 and 30000, which is the value for the Timer in milliseconds.

## Changes

### CAPI Syslog Messages

Syslog messages of the CAPI subsystem have been modified to be more informative now. CAPI now uses unique internal application identifications to make it easier to analyse debugging output.

Examples of new syslog messages:

#### **incoming call**

```
CAPI: DBG(34.023) APPL03:09 PLCI 0x0101 dialin from  
<> to local number <>  
CAPI: INF(34.040) APPL03:09 PLCI 0x0101 incoming  
call accepted
```

#### **outgoing call**

```
CAPI: INF(371.150) APPL04:1204 PLCI 0x2E01 dialout to <>  
CAPI: INF(371.172) APPL04:1204 PLCI 0x2E01 outgoing  
call established
```

In these examples, APPL04:1204 or APPL03:09 identify a unique CAPI application where the first number is an application ID and the second number an internal ID, which makes it easier to assign the syslog messages to one CAPI application.

### CAPI DATA\_B3\_IND message

CAPI DATA\_B3\_IND messages now contain a valid datablk counter.

Until now the datablk counter was unused and set to 0.

### Fax Storage: Forwarding

Concerns only BinGO! Professional.

When a CAPI application is started, faxes which are stored in the fax storage are now forwarded immediately to the application irrespective of the value of the variable *msgForwardDelay*.

## Fax Storage: Forwarding Order

Concerns only BinGO! Professional.

Faxes stored in the fax storage are now forwarded cyclically. When more than one fax is stored for the same recipient and a transmission error occurs, a different fax is forwarded on the next attempt.

## PABX: Idle Tone

Now two different idle tones are implemented to discern the type of dial procedure that is configured.

If the external prefix is set (external calls begin with a 0, internal calls do not have a special dial prefix), the idle tone is as follows: 3 short tones followed by a pause.

If the internal prefix “#” is set (internal calls begin with a #, external calls do not have a special dial prefix), the steady tone is the idle (dialing) tone.

## PABX: Configuration during an Established Call

To use the built-in telephony services of your router or to configure some ISDN features, you have to dial special codes starting with “\*”. Dialing may occur during an established connection. In such a case, to avoid the remote side also interpreting your DTMF tones as code, the connection is temporarily cut – the remote side does not receive any DTMF tones. The dialing procedure is as follows:

After dialing “\*”, the configuration mode is active and the connection is put on hold. Now you have to dial the second digit during the next 2 seconds, otherwise the configuration mode is left. After that dial the remaining digits.

## *Bugfixes*

### Fax Storage

The following bugfixes only concern BinGO! Professional.

- **Repeated Fax Forwarding**

When there is no more space in the fax storage, it can happen that an incoming fax can not be saved completely.

When, in such a case, the fax storage contained faxes which were tagged to delete, but were not yet deleted, the reception of a new fax was not interrupted immediately and sometimes an incorrect number of pages was written to the variable *msgDirDuration*. Consequently, the incorrectly saved fax was forwarded repeatedly and had to be deleted manually.

This bug has been fixed.

- **Acceptance of Incoming Calls**

When there was no more space in the fax storage and fax-polling was not configured, up to now incoming calls were accepted and after a short time canceled again.

Now no more calls are accepted in such a case, because any incoming call is a new fax and storing is not possible any longer, when there is no more space in the fax storage.

- **Reboot of the BinGO!**

When a fax was forwarded to a local CAPI application and this CAPI application interrupted the reception or the CAPI application itself was closed down, then there could occur a reboot of the BinGO! Professional.

This bug has been fixed.

### Reboot when Establishing ISDN Connections

- In rare cases a reboot of the BRICK occurred when outgoing ISDN connections were established. The typical output with such a kind of reboot was:

```
PANIC: MIB getnext
```

```
...
```

```
or
```

```
PANIC: kmem_free: unaligned pointer
```

```
...
```

This bug has been fixed.

## X.25 Routing Priorities

- The following problem occurred with X.25 connections from a BRICK across an ethernet link. When the link of the routing entry with lower metric (higher priority) was broken, the BRICK did not recognize it and nevertheless sent a CALL REQUEST to this address instead of selecting the route with the next higher metric.

This bug has been fixed by introducing a new timer in the variable ***L2SupervTime*** in the ***x25LinkPresetTable*** described under [New Timer in x25LinkPresetTable](#).

## NAT on a Dial-Up Interface

- When using NAT on a dial-up interface it could occur that no more sessions were allowed, also only few active NAT sessions were opened.

This bug could be recognized when the counter ***ipInAddrErrors*** was counted up and no more packets were routed, although the interface was up. The problem only occurred temporarily until one connection was disconnected.

This bug has been fixed.

## Setup Tool: WAN Partner Configuration

- There was a bug in Setup Tool when a number for ***both*** or ***incoming*** was configured for a WAN Partner. In such a case, the switch for **CLID** was falsely set to **no** by the sys-

tem, although numbers for **both** or **incoming** were adjusted. There also occurred problems when an **outgoing** number was configured in connection with certain encapsulations. The **CLID** switch then had to be set to **yes** to make it possible to save the configuration.

All bugs which occurred in connection with **CLID** and the configuration of **incoming** and **outgoing** numbers, as well as numbers with the direction **both**, have been fixed and Setup Tool is now working correctly.

### Dynamic IP Address Pools

- When the BRICK acts as a dynamic IP address server, IP addresses that are “reserved” for a certain connection partner are not assigned any longer when the respective IP address pool is moved or deleted.

### RADIUS OSPF Interfaces

- OSPF Interfaces (entries in the **ospfIfTable**) belonging to temporary RADIUS interfaces are now deleted after the RADIUS interface was closed down. This was done because the OSPF interfaces unnecessarily used up memory.

### Fax: T30 Carrier detection (V.21)

- This bugfix concerns only BinGO! Professional. With some fax devices a change of the modulation from data carrier to V.21 command carrier was not correctly recognized. The problem increased the number of faxes that were inadvertently disconnected.

This problem was fixed using another modem operating mode that causes the modem firmware to detect the modulation change.

### Accepting Calls with CAPI 1.1 Applications

- When an incoming call was accepted by a CAPI 1.1 application, the Called Party Number was returned automati-

cally as Connected Number to the caller. Some simple PABX could not handle this information and disconnected the call.

Now the Connected Number is not sent by CAPI 1.1 applications any longer.

### biboAdmCapiTcpPort/biboAdmTapiTcpPort

- To change the TCP port that is used for CAPI or TAPI applications one can configure the MIB variable ***biboAdmCapiTcpPort*** (***biboAdmTapiTcpPort***). This configuration can also be made via Setup Tool. If you wanted to apply the changes, you had to reboot the Brick.

Now the new value is used immediately after the modification and no reboot of the BRICK is necessary.

### IPX: ripCircTable and sapCircTable

- After the command `cmd=load` had been executed, the ***ripCircTable*** and ***sapCircTable*** contained each entry twice.

This bug has been fixed.

### IPX: Configuring the NetNumber

- When configuring a new WAN Partner using IPX, the NetNumber was reset to 0:0:0:0 and had to be corrected later manually.

This bug has been fixed.

### HTTP Server: Internet Explorer 4.0

- When trying to access the BRICK's HTTP server with Internet Explorer 4.0 to start applications like SNMP table browsing or "htmlshow", i.e. an application which required passwords to be started, user authentication failed. The reason was that the BRICK could not recognize the HTTP protocol version 1.1, which is used by Internet Ex-

plorer 4.0, and, therefore, assumed user authentication was not possible. These applications, however, require an authentication with a password and cannot be started without one.

This bug has been fixed by enhancing the HTTP server to be able to handle all current protocol versions.

A simple workaround to solve the problem if you do not update to this software release 4.9.4:

In the Internet Explorer 4.0's **View** menu, select **Internet Options...** and in the dialog box then access the **Advanced** Tab. Here you must clear the check box "Use HTTP 1.1".

### ISDN S<sub>0</sub>: Auto Configuration

- After a reboot it seldom occurred that a BinGO! Plus/Professional at a point-to-point connection needed too long, to detect the ISDN interface via auto configuration.

This bug has been fixed.

### Bridging

- **Compensation of Multiple Paths**

There was a problem with the spanning tree algorithm. In consequence of this, some ports were falsely set to **forwarding**, data packets were multiplied, and a high network traffic occurred. In rare cases the Brick rebooted.

This bug has been fixed.

- **dot1dStpPortPriority/dot1dStpPortPathCost**

To change the forwarding rules of the Brick you had to configure the MIB variables **dot1dStpPortPriority** and **dot1dStpPortPathCost**. These variables could not be adjusted however.

This bug has been fixed.



## Fax Applications with Protocol Switching

- This affected few fax applications with Protocol Switching with SELECT\_B\_PROTOCOL\_REQ. Two problems arose:
  - On sending a fax: BinGO! Plus/Professional did not confirm data transmission and the transmission was aborted by the application.
  - On receiving a fax: the sender confirmed correct transmission, although the application did not receive the fax data.

These bugs have been fixed.

## CAPI and Incorrect Bearer Capability

- Incoming CAPI calls, e.g. GSM calls, with incorrect bearer capability were not signalled to the CAPI. When the bearer capability contained additional bytes to that contained in the CAPI specification, no CIP value was recognised.

This bug has been fixed.

## TAPI: Calls Sent to TAPI Clients Several Times

- The PABX sent incoming calls to TAPI clients several times with different call IDs. Most Windows TAPI applications then either reported none or many calls instead of one call. In the latter case, when the TAPI application reported many calls, the user had to filter out the »right« call.

This bug has been fixed.

## PPP Callback Working on the 2nd Attempt

- PPP callback was known not to work under a combination of the following conditions:
  - on the 1st attempt after a reboot
  - with user-defined numbers
  - no entry on the *biboDialTable*

This bug has been fixed.

### Multiple Fax Forwarding with Different Extensions

- This bugfix concerns only BinGO! Professional.

If the fax memory was configured with several extension numbers and several faxes were simultaneously forwarded with these different extensions, it occurred, on rare occasions, that faxes for some extensions were not forwarded.

This bug has been fixed.

### Problems Accessing Compuserve for the First Time

- If the BRICK was newly configured, used as DHCP server on the LAN and you had configured Compuserve as your ISP, it was not possible to use a browser to establish a connection with your provider for the very first time.

In such a case, the router could not find a DNS server, necessary for connections using a name-based browser. Only connections to partners with PPP encapsulation were made and not to partners with x.75\_PPP or x.75\_BTXX\_PPP, as in the case of, for example, Compuserve access.

This bug has been fixed.

## Known Issues

### Faulty Reorganisation in Fax Memory

- This known issue concerns only BinGO! Professional.

In very few cases, incoming faxes can not be stored in the flash memory. The reorganisation of faxes in the memory does not occur properly when too many faxes marked »received« are still in the memory.

In such a case, you should either forward or delete some of the faxes in the memory. The reorganisation function then works and new incoming faxes can be stored.

### Connection Attempts between Client and Server

- When a client tries to connect with a server but the connection is denied, for example, as a result of the client not supporting MPPE, the state of the client interface does not turn to blocked, as it does not know the reason for the failure to connect. Continued attempts to establish the connection from client to server are made.

Should the connection attempt take place in reverse (client requests MPPE, server can not fulfil the request), the state of the interface on the client side turns to blocked.

### Deletion of Faxes from Fax Memory

- This known issue concerns only BinGO! Professional.

In rare cases it occurred that if several faxes are being read from the fax memory and if the receiving PC, or its CAPI, fails during the reception of a fax, the fax message receives the status **deleted** and does not stay in the state **received** in the fax memory.

### Outgoing FTP Connections via NAT

- When outgoing FTP connections occur via NAT, data transfer does not work with some FTP servers. The con-

nection is built up, the FTP client can register with the server. Commands such as `cd` and `pwd` work, but others such as `dir` and `get` do not.

The problem can be dealt with if the client is switched to the passive mode. This is not, however, possible with all FTP clients.

### Autologout Interrupting the Update

- If the autologout time interval is less than the time it takes to install the update to the flash, the autologout occurs, interrupting the installation of the image. The flash only has time to partially write the image to the flash ROM and the update is incomplete. The procedure must be repeated.

This can occur when a rather low time interval has been set. To remedy this, either set autologout to a higher value, i.e. a time interval longer than the time it takes to install the image, or with “`t 0`” disable autologout completely.

### Several Extensions for one POTS Port

- In general, one extension is configured for each POTS port.

For incoming calls, it is in principle possible to configure two or more extensions which are all directed to one POTS port. For outgoing calls, however, in such a case, anyone of these extensions is assigned as calling party number.

## *Detailed Feature Descriptions*

### Credits Based Accounting System

With dial-up WAN connections, it may occur that charges rise because of configuration errors. The Credits Based Accounting System gives BinGO! Plus/Professional administrators the ability to control charges. It allows the BinGO! Plus/Professional administrator to watch and limit the number of connections, the connection time and the accounted charges of every subsystem during a specified period of time. If the limit is exceeded the BinGO! Plus/Professional can not make further connections in that period of time. Syslog messages give you information about credits when the 90% or 100% mark for each limit and each subsystem is reached. Also, each time a call is rejected a syslog message is generated.

For the PABX subsystem, i.e. the POTS ports and the telephones connected to them, this feature also allows you to control incoming and outgoing connections. In this context, you should bear in mind that limiting incoming connections could result in not being able to accept incoming calls when your limit is reached.

When you adjust a maximum charge for outgoing calls of the POTS subsystem, you must consider that certain telephone companies do not transmit charging information, so that charges can not be counted.

Credits Based Accounting for the POTS subsystem will always take effect for both POTS ports. You can not configure credits for one single POTS port.

If a limit which is set for the POTS' outgoing calls is reached, you will hear the busy tone when lifting the handset. If the incoming calls' limit is reached, you also hear the engaged tone when trying to accept an incoming call, and the caller in this case first hears the ringing tone which changes into an engaged tone as soon as the called party tries to accept the call.

The new **isdnCreditsTable** controls this feature, it is described in the current MIB Reference at <http://www.bintec.de/download/brick/doku/mibref/index.html>.

The Credits Based Accounting System can also be configured via Setup Tool described below.

### Setup Tool Menus

In the Setup Tool main menu, there are two items containing menus for the Credits Based Accounting System: **ISDN** and **Monitoring and Debugging**.

**ISDN** With this new item, you can manage the Credits Based Accounting System.

**Monitoring and Debugging** Here you can find a new menu which enables you to monitor the incoming and outgoing connections and accounted charges.



```

BinGO! Plus Setup Tool                               BinTec Communications AG
[ISDN][CREDITS]: Configure Credits                    bingo
-----
Select Subsystem

Subsystem      Surveillance
capi           off
ppp            off
isdnlogin     off
pots          off

EXIT

Press <Ctrl-n>, <Ctrl-p> to scroll, <Return> to edit/select
    
```

Here you can see for which subsystems accounting is active (surveillance on) or inactive (surveillance off). The default value is off. To activate accounting for a subsystem, select the corresponding item and enter the detailed settings in the next step. There are four defined subsystems:

capi  
ppp  
isdnlogin  
pots

ISDN → Credits → ppp

BinGO! Plus Setup Tool [ISDN][CREDITS][EDIT]: Configure ppp Credits	BinTec Communications AG bingo
Surveillance	on
Measure Time (sec)	86400
Maximum Number of Incoming Connections	on 20
Maximum Number of Outgoing Connections	on 20
Maximum Charge	off
Maximum Time for Incoming Connections (sec)	on 28800
Maximum Time for Outgoing Connections (sec)	on 28800
SAVE	CANCEL
Use <Space> to select	

Here you can enter the detailed settings for the subsystem you have selected before, here, for example, ppp.

**Surveillance** = Determines whether or not accounting for ppp connections is activated. If you set Surveillance on, you are able to determine the following parameters.

**Measure Time (sec)** = The observation interval in seconds. Enter an integer from 0 to 2147483647. Default value is 86400 seconds, which is 24 hours.

**Maximum Number of Incoming Connections** = The number of allowed incoming connections during the measure time. If you set it on you can enter an integer from 0 to 2147483647. Default value is off.

**Maximum Number of Outgoing Connections** = The number of outgoing connections allowed during the measure time. If you

set it on, you can enter an integer from 0 to 2147483647. Default value is 100 calls.

**Maximum Charge** = The maximum charge information allowed during the measure time. If you set it on, you can enter an integer from 0 to 2147483647. Default value is off.

Charge information is measured in units or when charge information is sent as currency amounts, the charge is measured in 1/1000 of the respective currency. (E.g. receiving charging information "0.12 DM" would result in a value of 120 charging units.)

**Maximum Time for Incoming Connections (sec)** = The maximum allowed time in seconds for incoming connections during the measure time. If you set it on, you can enter an integer from 0 to 2147483647. Default value is 28800 seconds, which is 8 hours.

**Maximum Time for Outgoing Connections (sec)** = The maximum time allowed in seconds for outgoing connections during the measure time. If you set it on, you can enter an integer from 0 to 2147483647. Default value is 28800 seconds, which is 8 hours.



Monitoring and Debugging →

ISDN Credits →

BinGO! Plus Setup Tool [MONITOR][CREDITS]: Monitor Credits	BinTec Communications AG bingo
Select Subsystem	
Subsystem	Surveillance
capi	on
ppp	on
isdnlogin	on
pots	on
EXIT	
Press <Ctrl-n>, <Ctrl-p> to scroll, <Return> to edit/select	

Here you can see for which subsystems the Credits Based Accounting System is activated (surveillance on) or not activated (surveillance off). By selecting capi, ppp, isdnlogin or pots, you can check the remaining credits for each subsystem.



BinGO! Plus Setup Tool		BinTec Communications AG	
[MONITOR][CREDITS][STAT]: Monitor ppp Credits		bingo	
	Total	Maximum	% reached
Time till end of measure interval (sec)	7794	86400	91
Number of Incoming Connections	0	20	0
Number of Outgoing Connections	0	20	0
Time of Incoming Connections	0	28800	0
Time of Outgoing Connections	0	28800	0
Charge	0		
EXIT			

Here you can see the current values.

**Time till end of measure interval (sec)** = The seconds left in the current observation interval.

**Number of Incoming Connections** = The number of established incoming connections during the current measure time.

**Number of Outgoing Connections** = The number of established outgoing connections during the current measure time.

**Time of Incoming Connections** = The accounted time for incoming connections during the current measure time.

**Time of Outgoing Connections** = The accounted time for outgoing connections during the current measure time.

**Charge** = The number of charge information received during the current measure time.

Charge information is measured in units or when charge information is sent as currency amounts, the charge is measured in 1/1000 of the respective currency. (E.g. receiving charging information "0.12 DM" would result in a value of 120 charging units.)

# What Was New in Release 4.9.2

## Release 4.9 Revision 2

Released: 30.10.98

Features:

Bugfixes:

Detailed Description:

## Features

### New BRICKware for Windows

With Release 4.9 Revision 2, a new BRICKware for Windows (Release 4.9 Revision 1) is also available, which contains the Configuration Wizard.

BinGO! Plus/Professional now no longer has a separate version of BRICKware for Windows.



### New CAPI and TAPI Ports

BinTec product-specific TAPI and CAPI ports have been officially registered by the IANA (Internet Assigned Numbers Authority) and have been changed as follows:

	OLD PORTS	NEW PORTS
CAPI	6000	2662
TAPI	6001	2663

These default values are only used, when BinGO! and BRICKware are initially configured. It was necessary to introduce these changes, because, in rare cases, conflicts with applications which used old CAPI and TAPI ports occurred.

As a requirement for the operation of Remote CAPI/TAPI and the CAPI Tracer (PC), the values for the CAPI/TAPI ports configured on the BinGO! and the PC must be the same.

A software update on the BinGO! and on the PC does not change the configuration and with that also does not change the port numbers currently in use. Therefore, it is not necessary to change the ports after a mere update.

Nevertheless, we recommend using the new ports. In the long term, the new configuration will be necessary to resolve conflicts that may occur with NAT and Firewall configuration. Please note that incorrect configuration may be a potential source of errors.



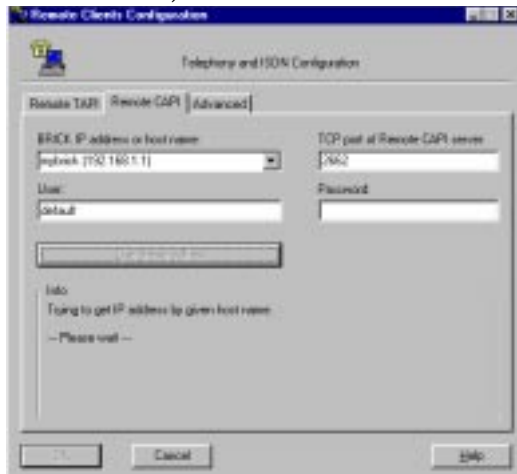
After configuring the CAPI/TAPI for the new port numbers, you must reboot the system for the changes to become effective. Also, ensure your changes have been saved to the boot configuration file using Setup Tool's Configuration Management menu, or using the "cmd=save" command from the SNMP shell. When only the BRICKware is newly installed or a new BinGO! is taken into operation or an old one is completely, newly installed, then the CAPI/TAPI ports must be manually adjusted. When BinGO! and BRICKware are completely, newly installed, no adjustments are necessary.

On the BinGO!, the CAPI port is configured in Setup Tool in the Static Settings menu of the IP menu, and the TAPI port in the Static Settings menu of the PABX menu.

BinGO! plus Setup Tool [IP][STATIC]: IP Static Settings	BinTec Communications AG bingoplus
Domain Name	bricks.com
Primary Domain Name Server	192.168.1.3
Secondary Domain Name Server	
Primary WINS	
Secondary WINS	
Time Protocol	TIME/UDP
Time Offset (sec)	0
Time Update Interval (sec)	86400
Time Server	192.168.1.3
Remote CAPI Server TCP Port	2662
Remote Trace Server TCP Port	7000
RIP UDP Port	520
BOOTP Relay Server	
Unique Source IP Address	
SAVE	CANCEL

BinGO! plus Setup Tool		BinTec Communications AG	
[PABX][STATIC]: PABX Static Settings		bingoplus	
Dial Procedure                      Prefix # for internal calls Remote TAPI Server Port          2663			
SAVE		CANCEL	

On the PC, the CAPI/TAPI server ports are configured in the program “Remote Clients Configuration”. The CAPI Tracer of the DIME Tools can be configured when starting a Trace session (Start/New CAPI Trace).



The current Unix Tools “capitrace”, “eft”, and “eftd” still use CAPI port 6000 as the default setting. The ports of these programs can be changed by setting the environment variable

“CAPI\_PORT” under Unix. (e.g : CAPI\_PORT=2662↵, export CAPI\_PORT↵)

## BinTec Configuration Wizard

With the Configuration Wizard (contained in BRICKware for Windows), BinTec offers you a convenient way to start running your router quickly and easily. You can make a basic configuration via the serial connection with your Windows PC. The basic configuration includes all important settings of the router and access to the Internet via an Internet Service Provider (ISP).

The Configuration Wizard guides you step by step through the configuration and so thorough knowledge of networking technologies is not necessary. Graphic illustrations and a context-sensitive online help system gives you additional support.

The software requirements for the Configuration Wizard are Windows 95 or later and Windows NT 4.0 or later.



Please note that the Configuration Wizard is only meant for an initial configuration of your router. It is not possible to change an existing configuration via the Configuration Wizard, because **all existing configurations are deleted** with a new configuration entered using the Wizard.

A detailed “Quick Install Guide” for the use of the new Configuration Wizard can be found with your product or can be retrieved from BinTec’s file server at <http://www.bintec.de> (Section: FTP Server) from your products FTP page.

## IP Route Announcement

In the *ipExtIfTable*, there is the new variable *ipExtIfRouteAnnounce*, which adjusts for each interface under which conditions, depending on the *ifOperStatus* (*ifTable*) of the respective interface, the routes defined on this interface are propagated.

This new variable is relevant for the routing protocols OSPF and RIP.

The variable can receive three possible values:

- ***up\_only***  
The routes are only propagated when the operational status of the interface is up.
- ***up\_dormant***  
The routes are only propagated when the operational status of the interface is up or dormant.
- ***always***  
Regardless of the operational status of the interface the routes are always propagated. If, for example, a dial-up interface is in the state “blocked”, the route is propagated.

In Setup Tool, the configuration is made under



BinGO! plus Setup Tool		BinTec Communications GmbH	
[WAN][ADD][IP][ADVANCED]: Advanced Settings ()		bingoplus	
RIP Send		none	
RIP Receive		none	
Van Jacobson Header Compression		off	
Dynamic Name Server Negotiation		yes	
IP Accounting		off	
Back Route Verify		off	
<b>Route Announce</b>		<b>up or dormant</b>	
Proxy Arp		off	
OK		CANCEL	
Use <Space> to select			

**up or dormant** specifies that the routes are only propagated when the operational status of the interface is up or dormant and is the default value.

**up\_only** specifies that the routes are only propagated when the operational status of the interface is up.

**always** specifies that independent of the operational status of the interface the routes are always propagated.

## Proxy ARP

The Proxy ARP (Address Resolution Protocol) is a technique to answer ARP requests for the hardware address of a particular IP address. Normally, ARP requests are answered by the station the IP address belongs to. With Proxy ARP the request can be alternatively answered by the BinGO!. This is useful when a host belonging to your local network is connected via WAN (e.g. a home office).

For a detailed description of the feature Proxy ARP, see BinTec's Software Reference, which is available via the WWW Server at <http://www.bintec.de> (Section: FTP Server) from your product's page.

With this software release the Proxy ARP feature has been enhanced. Proxy ARP must now also be configured on the destination WAN interface, via which the requested IP address would be routed.

For the LAN interface the variable ***ipExtIfProxyArp*** (***ipExtIfTable***) can receive the values off and on:

- ***off***  
Proxy ARP is turned off which is the default value.
- ***on***  
Proxy ARP is turned on.

In Setup Tool, Proxy ARP for the LAN can be configured in the Advanced Settings for the LAN interface.

For the WAN interface the variable ***ipExtIfProxyArp*** (***ipExtIfTable***) has been extended. When proxy ARP is turned on, ARP requests are answered depending on the ***ifOperStatus*** (***ifTable***) of the interface, via which the requested host can be reached. Possible values are ***off***, ***on*** and ***up\_only***.

Values for ***ipExtIfProxyArp*** on the WAN interface:



- **off**  
Proxy ARP is turned off which is the default value.
- **on**  
The request is only answered when the WAN interface has the **ifOperStatus up** or **dormant**. When the interface was in the state **dormant**, a connection is setup after the ARP request.
- **up\_only**  
The request is only answered when the WAN interface has the **ifOperStatus up**. This value makes sense when ARP requests are only to be answered if there is already an existing connection to the requested host.

In Setup Tool, Proxy ARP for the WAN interface can be configured in the WAN partner menu for the respective host in the Advanced Settings of the IP submenu.

The requirements for an answer to an ARP request from the LAN by the BinGO! are that the destination address be routed to a different interface from the LAN interface and that on both interfaces (LAN and destination WAN interface) proxy ARP is turned on (**on** for the LAN interface and **on** or **up\_only** for the respective WAN interface). Beyond that, the **ifOperStatus** of the WAN interface must have the required state.

When you want to use Proxy ARP on a RADIUS interface, the variable **ipExtIfProxyArp** must be set via the BinTec-specific RADIUS attributes. On using BinTec-specific RADIUS attributes, see the Extended Feature Reference available via the BinTec FTP server at <http://www.bintec.de>.



Because of the extension of the Proxy ARP configuration to the WAN interface, which means additional security, the old configurations made with prior software releases are no longer compatible. To reach the same functionality as before with an activated Proxy ARP on the LAN, the variable **ipExtIfProxyArp** must be set to **on** for the respective WAN interface.

## Access Lists

The range of values the variable ***ipFilterProtocol (ipFilterTable)*** can receive has been extended. The following protocols can additionally be defined for filtering: RSVP , GRE, ESP, AH, IGRP, L2TP. (For protocol descriptions see [http://www.iana.org/.](http://www.iana.org/))

In Setup Tool, the filters can be defined in the IP Access Lists menu.

## X.25 Dialout Without Configuration

In an X.25 network, there are often a lot of different connection partners that cannot all be configured on the BinGO! or even on different BinGO!s. In addition, there are often so many X. 25 partners that a configuration is not possible because of the limited size of the Flash ROM of the BinGO!.

For outgoing X. 25 calls a feature was implemented which generates an ISDN number out of the destination X.25 address or the destination NSAP.

For this feature, two new values for X.25 encapsulations have been added. The variable ***Encapsulation*** in the ***biboPPPT-able*** and the corresponding item ***Encapsulation*** in Setup Tool's WAN PARTNER/ADD menu can now also receive the value ***x25\_noconfig*** (Setup Tool: **X.25 No configuration**) and ***x25\_noconfig\_nosig*** (Setup Tool: **X.25 No configuration, No Signalling**).

The value ***x25\_noconfig*** uses X.25-specific signalling in the D-channel for the data call.

The value ***x25\_noconfig\_nosig*** is a variation of the value ***x25\_noconfig*** and uses in contrast to "X.25 No Configuration" ISDN-specific signalling in the D-channel for the data call.

A detailed description of "How do I configure X.25 dialout without configuration?" can be found in the current version of the Extended Features Reference, which can be retrieved from BinTec's FTP server at <http://www.bintec.de>.

## Pools for Dynamic IP Address Assignment

Beginning in software Release 4.9 Rev. 2, it is now possible to define separate IP address pools for dynamic IP address assignments. For Internet Service Providers (ISP) and other sites with many dial-in accounts, using IP address pools is convenient for defining separate user groups. One might assign “official” addresses from pool 1 for special accounts, and assign “non-official” addresses from pool 2 for private accounts.

At connect time, the BinGO! assigns an IP address from the Pool (Pool ID) defined for the respective WAN Partner. This Pool ID can be retrieved from,

1. the respective partner entry in the BinGO!'s ***biboPPPTable*** (using the new ***biboPPPIpPoolId*** variable),
2. a user record in the remote RADIUS server's users file with a BinTec-biboPPPTable="biboPPPIpPoolId=x" tag).

See the section [IP Address Pools](#) under Detailed Descriptions for additional information (including the updated Setup Tool menus).

## WINS (NBNS) Negotiation over PPP

The BinGO! now supports WINS (NBNS = NetBios Name Server) Negotiation over PPP.

A detailed description of this new feature can be found in BinTec's Software Reference in Chapter 7 under the heading “DNS and WINS (NBNS) Negotiation over PPP”. The Software Reference can be retrieved from BinTec's file server at <http://www.bintec.de> (Section: FTP Server). There you can find a link under “Reference Manuals” on the respective product page.

## DHCP Server Functionality

The DHCP server functionality of the BinGO! has been enhanced by the features DNS (Domain Name Server) and WINS (NBNS = NetBios Name Server) Relay.

A detailed description of this new feature can be found in BinTec's Software Reference in Chapter 7 under the heading

“DNS and WINS Relay”. The Software Reference can be retrieved from BinTec’s file server at <http://www.bintec.de> (section: FTP Server). There you can find a link under “Reference Manuals” on the respective product page.

## X.25 in Setup Tool

Two additional X.25 variables of the MIB now also can be configured via Setup Tool:



Now when you create a new configuration or edit a configuration in this menu, the item **L2 Window Size** can be configured for the respective Link. The default value is 2.

This item corresponds to the variable **L2WinSize** in the **x25LinkPresetTable**.



For each routing entry the item **Metric** can now be configured.

This item specifies a metric similar to the metric of an IP routing entry. If a call matches multiple entries in the x.25 Route Table, the routing entry with the lowest value of Metric will be used to route the call. The default value is 0.

The item corresponds to the variable **Metric** in the **X25RouteTable**.

## Changes

### TCP Optimization

TCP packets which are not confirmed are now repeated earlier. This speeds up the throughput for remote CAPI, remote TAPI and Telnet.

### Configuration: State File

When writing a state file with `cmd=state` the following variables are not output or substituted by “\*\*\*\*” with software release 4.9.2:

- ♦ All values of the variables of *bintecsec* are not output.
- ♦ The value of the variable *AuthSecret* of the *biboPPPTable* is substituted by “\*\*\*\*”.
- ♦ The value of the variable *Secret* of the *radiusServerTable* is substituted by “\*\*\*\*”.
- ♦ The value of the variable *Secret* of the *tafServerTable* is substituted by “\*\*\*\*”.

### CAPI: PLCI and NCCI

The internal process for building the values for PLCI (Physical Link Connection Identifier) and NCCI (Network Control Connection Identifier), which are used with connections between CAPI application and BinGO!, has been changed. Therefore, PLCI and NCCI are now not only unique for each application, but unique on each BinGO!.

### ipExtIfKeepalive

Up to now the maximum value for the variable *ipExtIfKeepalive* (*ipExtIfTable*), which defines the period between TAF short authentications, was 180 seconds. With a value higher than this “keepalive authentication” was performed continually.

Now the maximum value is 65535 seconds (about 18 hours). Above that, it is possible to turn off “keepalive authentication” by setting the variable *ipExtIfKeepalive* to 0.

### biboAdmBrdType

The value of the variable *Type* in the first entry of the *biboAdm-BoardTable*, which describes the type of BinGO!, has been extended. For the BinGO! now additionally the size of the Flash and of the CPU-DRAM (Central Processing Unit - Dynamic Random Access Memory) is stated.

Example for BinGO! Plus/Professional (2 MB Flash):

inx	Slot(*ro) PartNo(ro)	Type(ro) Connector(rw)	HWRelease(ro)	FWRelease(ro)
00	0 "BinGO! plus"	"BINGO PLUS (2/4 MB)" auto	"3.1"	"2.6"

In this example (2/4 MB) stands for: 2 MB flash and 4 MB CPU-DRAM.

### Charging Information

Because some PABX signal charging information in the D-channel in currency amounts, the registration of charging information on the BinGO! has been extended.

When charging information is sent as currency amounts, the charges can be read out of the variables *biboPPPConnCharge* and *biboPPPTotalCharge* in the *biboPPPStatTable* and the variable *biboPPPLinkCharge* in *biboPPPLinkTable*, where the charge is measured in 1/1000 of the respective currency. (E.g. receiving charging information “0.12 DM” would result in a stored value of 120 charging units.)



Please note that when charging information is sent as currency amounts, the feature Dynamic Shorthold is not available.

When charging information is sent as units, the charges can be read out of the variables *biboPPPConnUnits* and *biboPPP-*

**TotalUnits** in the **biboPPPStatTable** and the variable **biboPP-PLinkUnits** in **biboPPPLinkTable**.

The PPP accounting strings in the syslog messages (info level) have changed, too. Now charging amounts and charging units are output, where charging amounts are measured in 1/1000 of the respective currency (see above). Depending on which information is signalled, one of both variables is always set.

Example:

```
16:13:17 INFO/PPP: provider: outgoing connection closed, duration 21 sec, 10337 bytes received, 12235 bytes sent, 0 charging units, 120 charging amounts
```

## Bugfixes

### isdnLoginOnPPPDDispatch

- When the variable ***isdnLoginOnPPPDDispatch*** (***isdnTable***) is set to allow, incoming ISDN calls with the service indicator “telephony” should be routed to the ISDN login daemon, even though the call via the ***pabxExtensionExtension*** has the matching service “PPP” in the ***pabxExtensionTable***.

For products with modem hardware (FM-8MOD, CM-2XBRI,...) it happened that incoming calls with this signalization were dispatched to the PPP routing, so that no login was possible.

This bug has been fixed.

### LAPB Encapsulation with Compression

- Especially for leased line connections it occurred that with LAPB encapsulation (IP\_LAPB resp. MPR\_LAPB) and compression (V. 42bis) data transfer was not possible. The reason was an inconsistency of the compression and decompression histories, which could result from a layer 1 disconnect. In spite of this failure the value of the variable ***ifOperStatus*** (***ifTable***) was remaining “up”.

This bug has been fixed and those inconsistencies should not occur anymore.

### biboPPPLQMTTable

- For dial-in connections with inband authentication the interface index was not set in the ***biboPPPLQMTTable***, when PPP Link Quality Monitoring was negotiated.

This bug has been fixed.

### B-Channel Bundling with RADIUS

- When a BinGO! was receiving more than 32 calls from dial-in partners via RADIUS at the same time, from the 33



dial-in on, partners were assigned wrong interface indexes (between 10000 and 15000; correctly the interface indexes for RADIUS PPP start at 15001). When one of these partners with wrong indexes, then called again for a B-channel bundling, it resulted in a data transfer failure, because it could not be recognized that there are already existing connections for this partner.

This bug has been fixed.

## CAPI

- CAPI User Login for BinGO! Plus/Professional

Applications using old drivers, which do not perform a user login, could dial out, although the user “default” was not configured in the *pabxUserTable* or a password was configured for the user “default”.

This bug has been fixed.

- CAPI User Login for BinGO! Plus/Professional

This concerns only applications using new drivers and performing a user login, when starting the program. After 256 correctly performed user logins the BinGO! Plus/Professional rebooted. This meant that (without a reboot) an application could only be started 256 times.

This bug has been fixed.

- Direct-Dial-In with CAPI Applications

To receive the whole Called Party Number of an incoming call at a point-to-point ISDN interface, a CAPI application has to collect the information out of several CAPI messages it receives from the BinGO!.

For this purpose some applications only interpret those digits, which are signalled to them with the “INFO\_IND” message and ignore the digits in the “CONNECT\_IND” message. These digits of the “CONNECT\_IND” message were not sent additionally in an “INFO\_IND” to the application.

Especially when the Called Party Number was received completely in one message by the BinGO!, the application did not get any "INFO\_IND" message and it occurred that in such a case a call was incorrectly accepted or not accepted.

This bug has been fixed.

Now all digits of the Called Party Number, which are contained in the "CONNECT\_IND" message are additionally sent in an "INFO\_IND" message.

- **Data Transfer in Transparent Mode**

When a CAPI application was sending data using the B-channel in transparent mode, it sometimes occurred that at the end of a transmission up to 31 byte were lost.

This bug has been fixed.

## **Network Address Translation**

- After receiving several broadcast packets via an interface where NAT is being performed the BinGO! either "locked-up" or inadvertently rebooted. If the system locked up the BinGO! was no longer accessible (via remote or console) and had to be switched on and off.

This bug has been fixed.

## **RAS and Remote TAPI (BRICKware)**

- There occurred the problem that when RAS (or other services) were installed under Windows NT, the TAPI could not be started. This happened because the remote TAPI wrote its configuration data to a user-specific part in the registry. When booting and before a user logged in the PC tried to start TAPI and to read the configuration out of the default part of the registry. Therefore the TAPI could not be started.

This bug has been fixed in the current Release of the BRICKware for Windows. Please notice that after the installation of the new BRICKware you will have to reboot

your PC twice until this change in the software is activated.

### localUdpAllowTable

- When the ***localUdpAllowTable*** contained more entries than the ***localTcpAllowTable***, there sometimes occurred a reboot of the BinGO!.

This bug has been fixed.

### CM-2XBRI and CM-1BRI/2BRI Modules

- The BRI modules stopped reception, when they received very large data packets via the B-channel. This bug occurred with installations enclosing IPX and Token Ring. It is possible that this bug also appeared with CAPI applications on the BIANCA/BRICK XS/XS office, the BinGO! and the BinGO! Plus/Professional.

The bug has been fixed.

### Call Collisions with MS Callback

- Microsoft Windows clients only accept incoming calls, when before, via CBCP, a callback was negotiated. Sometimes a dial-out to these clients was conducted, which was not negotiated as described and a call collision occurred, which could cause that the MS Callback was not successful.

This bug has been fixed.

### ifconfig Command

- It was not possible to use the “ifconfig” command on a completely unconfigured BinGO! to set the BinGO!’s IP address on the LAN.

This bug has been fixed.

Now you can use e.g.  
ifconfig en1 168.1.1.1 netmask 255.255.255.0 up  
to configure the IP address.

### Setup Tool: WAN Partner

- When configuring a new WAN partner and with that setting **IP Accounting** in the [WAN][ADD][ADVANCED] menu to **on**, the **IP Accounting** value was reset to **off**, although the menu was left with **Save**.

This bug has been fixed.

### Setup Tool: Access Lists

- On a BRICK with access lists for more than 100 interfaces configured using the Setup Tool, there sometimes occurred a robot.

This bug has been fixed.

### DTMF Tones during Telephone Calls

- When operating BinGO! Plus/Professional it sometimes occurred that during telephone calls certain frequency ranges of human voice were incorrectly recognized as DTMF tone input.

This bug has been fixed.

The duration to detect a DTMF tone has been extended from 25 ms to 45 ms.

### Fax Storage: Incoming Calls

- This bugfix concerns the BinGO! Professional only. Incoming calls with the service indicator “telephony” instead of “facsimile group 2/3” were not accepted by the fax storage.

This bug has been fixed. Now calls from all analog remote stations are accepted.

## Known Bugs

- B-Channel Trace

When a B-channel was traced (via the command “trace”), the throughput of this B-channel is no longer taken into account after the trace is stopped. The result is that for the dynamic channel-bundling a B-channel may be closed down too early or no additional B-channel is opened, although there is enough load on the connected channels.

- ISDN Causes in Accounting Messages

In the current software release the output of the ISDN causes in accounting messages is not yet implemented.

- Fax Storage (BinGO! Professional only)

When a fax message could not be saved completely, because there is not enough free memory in flash ROM, it is still checked for completeness, when it is forwarded or polled. Because it will never be complete, forwarding is repeated infinitely and for polling this means that the fax message can be polled repeatedly.

We recommend to delete incomplete fax messages manually from the fax storage via the Setup Tool (Fax Storage/Memory menu).

## Detailed Feature Descriptions

### IP Address Pools

#### Pool ID Selection

When dynamically assigning an IP address to a dial-in client the address, which will be assigned respectively the Pool, from which the address is retrieved is determined in the following order.

#### 1. Assigning a Static IP Address

When there exists an entry in the **ipRouteTable** for the dial-in client, where **ipRouteMask** is set to a host route (= **255.255.255.255**) and **ipRouteType** has the value **direct**, in this case the IP address stored in the variable **ipRouteDest** of this routing entry is taken to be assigned for this WAN partner.

If caller can't be authenticated locally via the MIB, RADIUS server(s) are contacted. If a server authenticates the caller, and there is a User-Record entry

```
BinTec-ipRouteTable="ipRouteMask=255.255.255.255
                    ipRouteType=direct
                    ipRouteDest= x"
```

the IP address stored in the variable **ipRouteDest** of this entry is taken to be assigned for this WAN partner.

#### 2. Assigning an IP Address from an Address Pool

When the procedures described under 1. was not successful, the IP address is assigned from the Pools.

Once the caller is identified (either inband or outband), the WAN partner's **biboPPPTable** entry is compared. If the **IPAddress** field = "dynamic\_server" AND an address is available from the pool identified by the **PoolId** field, then a free address is assigned.

If caller can't be authenticated locally via the MIB, RADIUS server(s) are contacted. If a server authenticates the caller and there is a User-Record entry BinTec-biboPPPTable="biboPPPIpAddress=dynamic\_server", the pool ID is deter-

mined from the User-Record entry BinTec-biboPPPTa-  
ble="biboPPPIpPoolId=x".

## MIB Tables Overview

Overview of new/updated system tables used in conjunction with Address Pools for dynamic IP address assignment.

*Updated!* **biboPPPTable**

Main system table for partner-specific PPP settings. Updated to include **IpPoolId** variable.

*Updated!* **biboPPPIpAssignTable**

Contains ranges of IP addresses that make up one or more logical Address Pools. Updated to include **PoolId** and **Range** variables.

*New!* **biboPPPIInUseTable**

Contains entries for each address that is currently assigned/reserved. The BinGO! updates the entries dynamically via the **State** field.

For detailed description of individual system table fields please refer to the BIANCA/BRICK MIB Reference on the accompanying Companion CD or at [BinTec's WWW](#) site.

## Example Configuration of IP Address Pools via Setup Tool

### A. Dial-In Partner without RADIUS

**IP** → **DYNAMIC IP ADDRESS** → **ADD**    Create Address Pool

First, create/modify a Pool ID to contain IP addresses that will be available for assignment at connect time.

Pool ID	1
Number	10.5.5.5
Number of Consecutive Addresses	5

WAN PARTNER → ADD

### Create Partner Interface

Here you'll need to set:

Partner Name	test
Encapsulation	PPP
Compression	none
Encryption	none
Calling Line ID	no

Then, in the IP submenu configure the BinGO! as a Dynamic IP Address server for this partner.

IP Transit Network	dynamic_server
--------------------	----------------

In the ADVANCED SETTINGS submenu define the Pool ID

IP Address Pool	1
-----------------	---

#### B. Dial-In Partner with RADIUS server

IP → DYNAMIC IP ADDRESS → ADD Create Address Pool

Next, modify a Pool ID to contain IP addresses that will be available for assignment at connect time.

Pool ID	2
Number	192.168.80.20
Number of Consecutive Addresses	20

Then you must define the following entry in the User-Record of the RADIUS server:

BinTec-biboPPPTable="biboPPPIpPoolId=2"

#### Example Configuration of IP Address Pools via SNMP Shell

In the following examples the SNMP shell input shown in the examples A.1, A.2, and B.1 must be entered in one command line.

##### A. Dial-In Partner without RADIUS

###### 1. Create an IP address pool in the *biboPPPIpAssignTable*.

```
bingo:> biboPPPIpAssignAddress=10.5.5.5
biboPPPIpAssignPoolId=1
biboPPPIpAssignRange=5
```



2. Set the WAN partner in **biboPPPTable** to use Pool ID. Assuming entry 4 is the existing WAN partner we want to configure for Dynamic IP address assignment

```
bingo:> biboPPPIpPoolId:4=1
biboPPPIpAddress:4=dynamic_server
```

## B. Dial-In Partner with RADIUS server

1. Create an IP Address pool in the **biboPPPIpAssignTable**.

```
bingo:> biboPPPIpAssignAddress=192.168.80.20
biboPPPIpAssignPoolId=2
biboPPPIpAssignRange=20
```

2. Define the following entry in the User-record of the RADIUS server:

```
BinTec-biboPPPTable="biboPPPIpPoolId=2"
```

3. Once the caller is authenticated via a RADIUS server a temporary **biboPPPTable** entry is generated. The **PoolId** field for this entry is determined by the contents of the User-Record discussed above.

## Important Note:

### Overlapping Address Pools:

Although it's legally possible to define IP address pools that overlap (as shown below), the BinGO! will not assign an address twice.

The **biboIpInUseTable** is consulted for this purpose.

inx	Address(*rw)	State(-rw)	PoolId(rw)	Range(rw)
0	10.5.5.1	unused	0	2
1	10.5.5.2	unused	1	2
2	10.5.5.3	unused	2	2

With the **biboPPPIpAssignTable** shown above, only four IP addresses could actually be used at any given time.