http://www.bintec.de

RELEASE NOTE

BinGO! Plus
BinGO! Professional

September 21, 1998

New System Software:
Release 4.8 Revision 7

This document describes the new features, enhancements, bugfixes, and changes to the BinGO! Plus/BinGO! Professional System Software since Release 4.8 Revision 3. Please, take notice of the New Update Procedure described on page 6.

Contents:

Upgrading System Software

1. Retrieve the current system software image from BinTec's WWW server at http://www.bintec.de (Link: 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) 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 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, there is the option **-v** that verifies the image file. For more detailed information see New Update Procedure on page 6.
4. Once you've installed Release 4.8 Revision 7 you may want to retrieve the latest documentation (in Adobe's PDF format), which is also available from BinTec's FTP server at the address noted above.

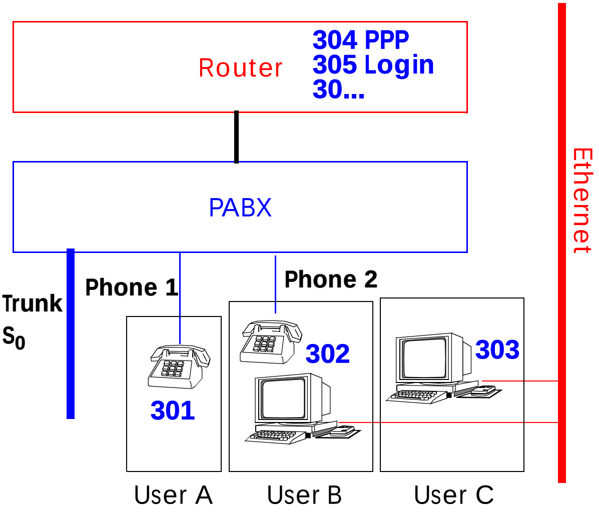
   Note: When upgrading system software, it is also recommended that you use the most current versions of *BRICKware for Windows for BinGO! Plus/Professional* and *UNIXTools*. Both can be retrieved from BinTec's FTP server.

The Voice Data Product Line

The products BinGO! Plus and BinGO! Professional belong to the BinTec product line that was enhanced by the PABX functionality. This concept means integrating voice into the router product and is a further step to “Integrated Services Networking”.

Combining the router functions with PABX allows an easy and cost-effective implementation of many new applications like e.g. Computer Telephony Integration (CTI).

The PABX part of the device is connected to the ISDN network and on the other hand to different terminals (phones, computer applicatons, ...).



To meet security necessities a user concept is introduced together with the PABX concept. This user concept includes that extension numbers are related to single users and also terminals are configured for the respective users.

What's New in Release 4.8.7

Release 4.8 Revision 7:

Released: 21.09.98

Features: | Bugfixes: | Detailed Description:

Features

New BRICKware for BinGO! Plus/Professional

Since Release 4.8.5 there is a new separate *BRICKware for BinGO! Plus/Professional*, which in its current version 4.8.5 is included on the BinGO! Plus / Professional Companion CD respectively can be retrieved from BinTec's WWW server at http://www.bintec.de (Link: FTP-Server).

Setup Tool Menu Reorganization

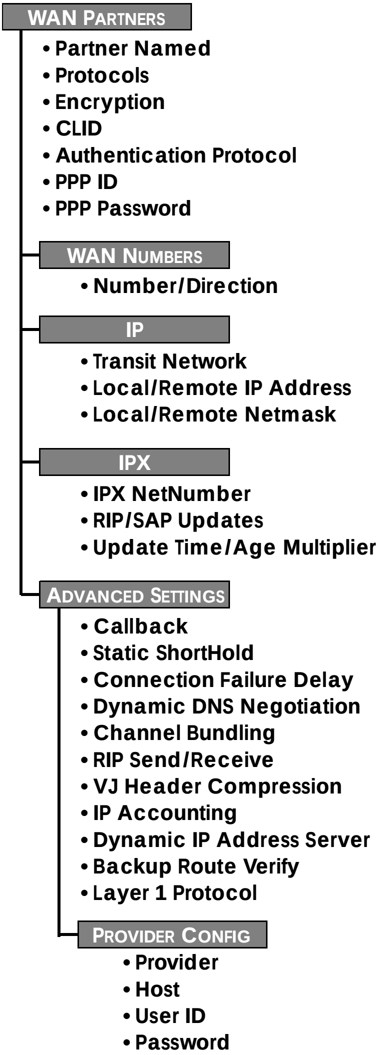
Beginning in Release 4.8 Revision 7, Setup Tool's Partner Management menus (contained in the WAN Partners section) have been reorganized to reflect a more logical structure. With the exception of several new configuration options (described below) these changes only affect the menu structure. Protocol specific configuration settings in Setup Tool's WAN Partners menu have been moved to the new / updated PPP , IP , IPX , and BRIDGE submenus.

As a guide to the new menu structure, the diagram shown on the following page shows both the old and the new menu layouts. The descriptions of existing configuration fields can be found in the most current version of the BinGO! Plus / Professional User's Guide which is available via BinTec's WWW Server.

Also note that the new menu structure outlined below is also reflected in Setup Tool's X.25 → MPR → , FR → MPR → , and VPN → submenus.

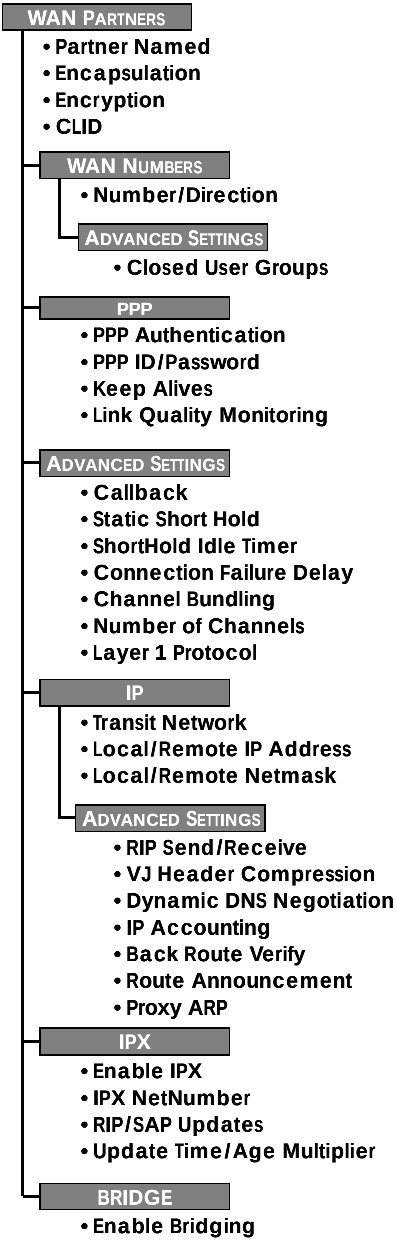
Detailed information about new configuration options appearing in this release can be found in the section Detailed Feature Descriptions on page 12.

Old Setup Tool WAN Partner Menus



- WAN PARTNERS
  - Partner Named
  - Protocols
  - Encryption
  - CLID
  - Authentication Protocol
  - PPP ID
  - PPP Password
  - WAN NUMBERS
    - Number/Direction
  - IP
    - Transit Network
    - Local/Remote IP Address
    - Local/Remote Netmask
  - IPX
    - IPX NetNumber
    - RIP/SAP Updates
    - Update Time/Age Multiplier
  - ADVANCED SETTINGS
    - Callback
    - Static ShortHold
    - Connection Failure Delay
    - Dynamic DNS Negotiation
    - Channel Bundling
    - RIP Send/Receive
    - VJ Header Compression
    - IP Accounting
    - Dynamic IP Address Server
    - Backup Route Verify
    - Layer 1 Protocol
    - PROVIDER CONFIG
      - Provider
      - Host
      - User ID
      - Password

New Setup Tool WAN Partner Menus



- WAN PARTNERS
  - Partner Named
  - Encapsulation
  - Encryption
  - CLID
  - WAN NUMBERS
    - Number/Direction
    - ADVANCED SETTINGS
      - Closed User Groups
  - PPP
    - PPP Authentication
    - PPP ID/Password
    - Keep Alives
    - Link Quality Monitoring
  - ADVANCED SETTINGS
    - Callback
    - Static Short Hold
    - ShortHold Idle Timer
    - Connection Failure Delay
    - Channel Bundling
    - Number of Channels
    - Layer 1 Protocol
  - IP
    - Transit Network
    - Local/Remote IP Address
    - Local/Remote Netmask
    - ADVANCED SETTINGS
      - RIP Send/Receive
      - VJ Header Compression
      - Dynamic DNS Negotiation
      - IP Accounting
      - Back Route Verify
      - Route Announcement
      - Proxy ARP
  - IPX
    - Enable IPX
    - IPX NetNumber
    - RIP/SAP Updates
    - Update Time/Age Multiplier
  - BRIDGE
    - Enable Bridging

Partner-Specific/non-Specific PPP Settings

In Setup Tool partner specific PPP settings can be configured under the new WAN Partners → PPP → menu. For information regarding the available partner-specific settings, see New Partner-Specific PPP Settings on page 13.

Default PPP settings (partner non-specific) can also be configured via the new PPP → menu. For information regarding these settings refer to Global PPP Settings Menu on page 12.

New Update Procedure

Performing a software update on a running system via the **update** command (SNMP shell) requires that a contiguous block of free memory, greater than or equal to the size of the new software image, is available. In the past there occured problems for BRICKs with 4 MB RAM, when the update application could not allocate enough memory to load a software image into RAM via TFTP.

With Release 4.8.7 the Update procedure was enhanced so that also in this case an update via the **update** command is possible.

When there is not enough memory available to load the complete image into RAM, the user is offered an incremental update. Then the new software image is loaded in blocks of 64 KB via TFTP and written to Flash ROM immediately. Because this procedure offers no possibility to check the integrity of the image, there is the option **-v** that verifies the image file.

Note that independent from software image size and available RAM you can always perform an update using the BOOT-monitor.

In the following an example for the new update procedure (verifying the image file and updating the BRICK):

1. Verifying the image

```
bingo:> update -v tftpserver bgo_487.bgp
Starting File Transfer....................... OK (754)
Checking new image... OK
File verifies OK
```

2. Updating the BinGO! Plus / Professional with the image

```
bingo:> update tftpserver bgo_487.bgp
Starting File Transfer .
Your current software release is 4.8.6.
New image has release 4.8.7.

WARNING: There is not enough free memory (RAM) to store
the new software image before writing it to flash. You
can perform an incremental update (the image is written
directly to flash in 64 KB increments). If you
need to perform an incremental update you should restart
the update using the -v option to verify the integrity
of the new file.
Don't reboot the router during the update.

Do you want to perform an incremental update (y or n)
[n] ? y
Receiving and Writing to Flash ROM ............
Software update complete
Reboot now (y or n) [n] ? y
```

We recommend to first verify the software image file and to start the incremental update after a sucessful verification of the file.

Transfering Configuration Files via the Serial Port

With Software Release 4.8.7 it is possible to load and save configuration files via the serial interface using the protocol XMODEM (up to now only possible via TFTP). Therefore the variable ***file*** is assigned the value **xmodem** or **xmodem-1k**. **xmodem-1k** uses a packet size of 1024 Byte (default: 128 Byte) and in general reaches a higher throughput. The packet size is defined by the sender so that the value **xmodem-1k** only makes sense on the sending end; on the receiving end it is ignored.

To make use of this new feature you have to access your BinGO! Plus / Professional from a computer via the serial port and a terminal program as described in “Getting Started” in the Chapter “Configuration” (“Over Serial Port”).

Getting the Configuration

```
cmd=get file=xmodem path=new_config
```

loads a file received via XMODEM with the name new_config into Flash ROM.

After this command has been started the terminal program must be set to Send (Upload) and the transmission protocol (XMODEM) as well as the source file name and location must be entered. For the time of the file transfer the console cannot be used.

Putting the Configuration

```
cmd=put file=xmodem path=boot
```

sends the Flash ROM file boot via XMODEM.

After this command has been started the terminal program must be set to Receive (Download) and the transmission protocol (XMODEM) as well as the destination file name and location must be entered. For the time of the file transfer the console cannot be used.

Transmitting State Information

The previously mentioned commands only send or retrieve the configuration files containing variables with Read-Write status. They send / retrieve information from files stored in flash. Using “cmd=state”, you can save all configuration information currently in memory. This information includes Read-Write AND Read-Only data such as status / accounting information and passwords.

```
cmd=state file=xmodem
```

Configuration files may contain the current passwords for the Read, Write and Admin Communities. If you use the cmd=put or cmd=state to transfer BinGO! configuration files, you should also control access to these files for security reasons.

When nothing is specified the currently selected baud rate is used for the transfer. The transfer baud rate can be changed by adding @baud to the file variable, e.g.:

```
cmd=put file=xmodem@9600 path=boot
```

Possible baud rates are 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200. For transmitting data to the BinGO! (cmd=get) you should not select a rate higher than 9600. Selecting higher than default baud rates may result in transmission errors.

In case of transmission errors a syslog is generated.

This feature can only be used via the SNMP shell, not via Setup Tool.

Fax Storage

The fax storage feature is only available on BinGO! Professional.

The fax storage can receive faxes on extensions (MSNs) assigned to it and store them in the Flash ROM of the BinGO! Professional. The fax storage has a capacity of 0.75 MB, and this (depending on FAX contents) corresponds to about 20 fax pages.

Faxes stored in the Flash can be forwarded to a configured phone number and/or can be called up by faxpolling. When a fax is forwarded, the forwarding is repeated with a adjustable forwarding delay until the forward is successful.

The standard usage of the fax storage will be to use the BinGO! Professional to store faxes for the time your computer is turned off and with that the CAPI application (e.g RVS-COM lite) is not available. This means that for the MSN, for which incoming voice (or voice and data calls) are normally routed to the CAPI application, you must create an entry in the Fax Storage Extension menu, which will forward the faxes to the CAPI application of your computer as soon as it is available again. Make sure that the extensions configured for the respective CAPI application and the fax storage are the same. Then you can also decide, whether you want to allow faxpolling for stored faxes.

Faxpolling

There is the possibility to poll faxes from the fax memory using a remote fax. The number that must be used for faxpolling is the phone number with the extension configured for the fax storage.

Updating To a New Software Image

Before Updating to a new software image the fax storage must be read out (by forwarding or faxpolling) and deleted, because depending on the size of the new software image the fax storage could be overwritten.

Detailed information about the configuration of the fax memory functionalities you can find in the section Detailed Feature Descriptions on page 12.

Transparent Data Connections via POTS Ports

The BinGO! Plus / Professional now supports transparent data connections between local POTS ports and remote CAPI applications.

Bugfixes

ISDN Autoconfiguration

- Existing values for the BinGO!'s Country Code, Area Code, and Subscriber Number (configured in the ***pabxTrunkTable***, or the WAN Interface menu) were incorrectly deleted at boot time when the **ISDN Autoconfiguration** option was set to “autodetect on bootup”.
  This problem has been corrected in Revision 7.

Modem

- When a modem connection was active on the BinGO! Plus a second data connection sometimes attempted to access the same modem and resulted in a system reboot. This problem has been corrected.

X.25

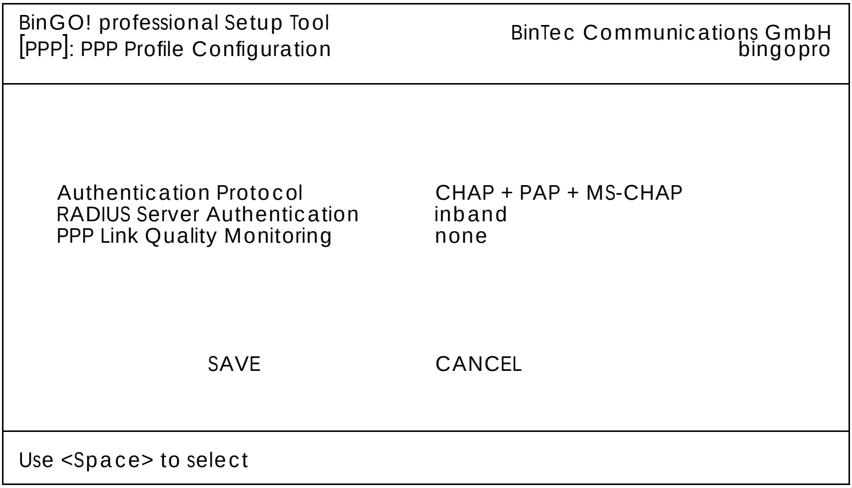
- Packet / Window Size facilities are now transmitted in CALL CONF packets on Layer 3 connections when the respective facility is requested / negotiated in the Call packet. In previous releases these facilities were only transmitted when the negotiated values did not reflect the requested values.
- In connection with RFC 1086 TP0 connections the following problem has been corrected. Under certain circumstances RFC 1086 data packets were signalled with a length of 0 bytes for Protocol ID and CallUserData fields but actually contained 4 null-bytes for CallUserData. This has been corrected. The CallUserData field is now (correctly) empty in such cases.

Detailed Feature Descriptions

New Setup Tool PPP Configuration Opions

Global PPP Settings Menu

The new **PPP** → menu has been added to Setup Tool's main menu to allow you to configure default (non-partner specific) PPP settings. The PPP settings configured in this menu are only used when negotiating an incoming call that could not be initially identified via Calling Line ID.



```
BinGO! professional Setup Tool
[PPP]: PPP Profile Configuration               BinTec Communications GmbH
                                                                  bingopro


   Authentication Protocol            CHAP + PAP + MS-CHAP
   RADIUS Server Authentication       inband
   PPP Link Quality Monitoring        none


               SAVE                   CANCEL

Use <Space> to select
```

The possible "default" PPP settings available in this menu include:

Authentication Protocol = Defines the type of PPP authentication protocol to offer the caller first. Possible values include: none, PAP, CHAP, CHAP + PAP, MS-CHAP, and CHAP + PAP + MS-CHAP.

RADIUS Server Authentication = This entry is used to configure possible RADIUS authentication on incoming calls. When set to "inband" (the default) only inband RADIUS requests (PAP, CHAP) are sent to the defined RADIUS server. When set to "Calling Line ID" outband requests are sent to the server. When set to “both”, both requests are sent. Setting to “none” disables RADIUS requests.

PPP Link Quality Monitoring = Defines whether link quality monitoring is performed for PPP links. When set to “yes”, link statistics are written to the ***biboPPPLQMTable***. For detailed information about Link Quality Monitoring see Link Quality Monitoring page 36.

New Partner-Specific PPP Settings

Two new options, PPP Keep Alives and Link Quality Monitoring, have been added to the WAN Partners → PPP → submenu.

PPP Keep Alives = When this option is set the BinGO! sends LCP echo requests to the remote partner every three seconds. After five unanswered requests the PPP interface’s ***ifOperStatus*** is set to “down”. PPP keep alives is most useful (and by default, set to “on”) for leased line interfaces. The transmission of echo requests does not affect the Short Hold timer.

Link Quality Management = This option allows you to tell the BinGO! to gather PPP Link Quality statistics for a specific PPP partner. When enabled, link statistics are written to the BinGO’s ***biboPPPLQMTable*** each time a connection is established with this partner. For detailed information about Link Quality Monitoring see Link Quality Monitoring page 36.

Route Announcement

The Configuration option *Route Announce* has been added to the WAN Partners → IP → ADVANCED SETTINGS → menu. This option allows you to define under what conditions known IP routes are advertised (via the RIP or OSPF routing protocols) by the BinGO! Plus / Professional. The following three settings can be defined via this option.

- Route Announce = up_only
  When set to “up_only” an IP route will only be advertised

if the operational status (value of ***ipOperStatus***) of the interface (to which the route points to) is up.

- Route Announce = up_dormant
  When set to “up_dormant” an IP route will only be advertised if the operational status of the interface (to which the route points to) is either up or dormant.

- Route Announce = always
  When set to “always” the IP route will always be advertised, regardless of the interface’s ***ifOperStatus***.

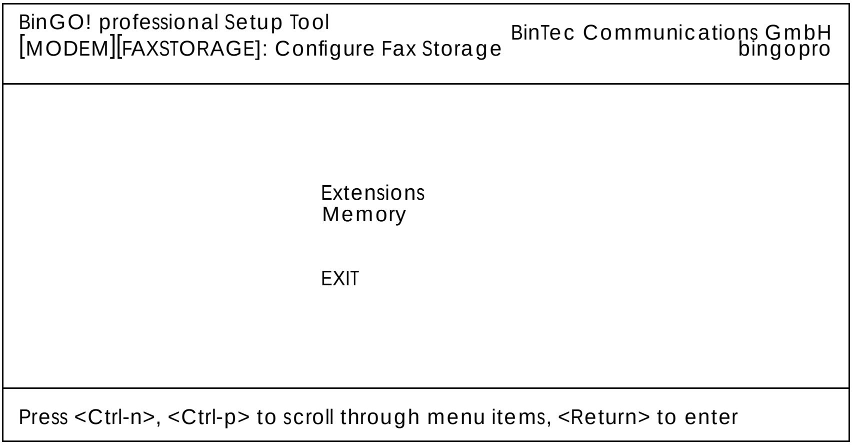
Fax Storage

Together with the fax storage two new tables are introduced into the MIB for the configuration of this feature. Settings can also be made via the Setup Tool and the new submenus.

Setup Tool

In Setup Tool’s MODEM → you will find a new **Fax Storage** submenu.

MODEM → FAX STORAGE →



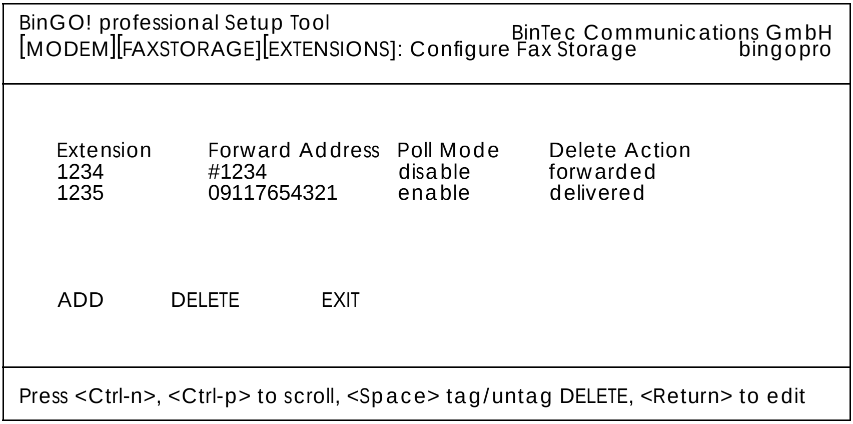
```
BinGO! professional Setup Tool                        BinTec Communications GmbH
[MODEM][FAXSTORAGE]: Configure Fax Storage                               bingopro




                              Extensions
                              Memory

                              EXIT




Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter
```

The EXTENSIONS → submenu serves to configure faxforwarding and contains items from the msgForwardTable (described below).

The MEMORY → submenu displays the contents of fax memory and contains items from the msgDirTable (described below).

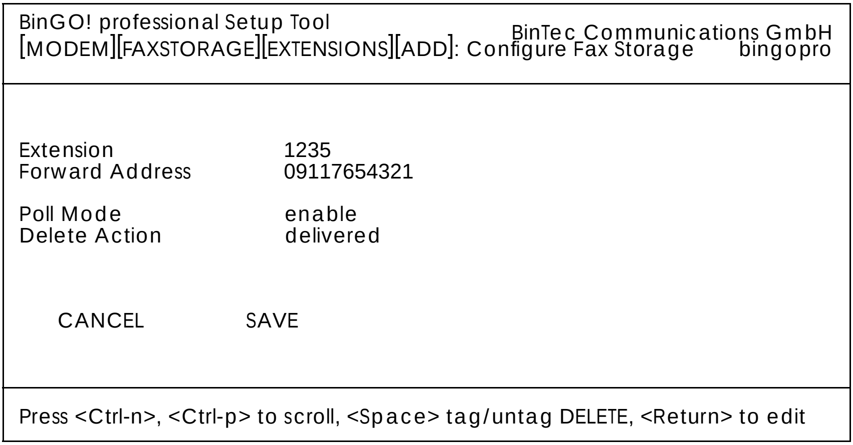
MODEM → FAX STORAGE → EXTENSIONS →

Here you can see entries already made and you can add, delete and edit entries.



```
BinGO! professional Setup Tool                              BinTec Communications GmbH
[MODEM][FAXSTORAGE][EXTENSIONS]: Configure Fax Storage                        bingopro


    Extension        Forward Address   Poll Mode       Delete Action
    1234             #1234             disable         forwarded
    1235             09117654321       enable          delivered


    ADD          DELETE          EXIT


Press <Ctrl-n>, <Ctrl-p> to scroll, <Space> tag/untag DELETE, <Return> to edit
```

MODEM → FAX STORAGE → EXTENSIONS → ADD →



```
BinGO! professional Setup Tool                              BinTec Communications GmbH
[MODEM][FAXSTORAGE][EXTENSIONS][ADD]: Configure Fax Storage                   bingopro


Extension                    1235
Forward Address              09117654321

Poll Mode                    enable
Delete Action                delivered


    CANCEL          SAVE


Press <Ctrl-n>, <Ctrl-p> to scroll, <Space> tag/untag DELETE, <Return> to edit
```

Extension = The extension that is used to accept fax calls for the fax storage. For this extension there must exist an entry in the **PABX/Extensions** submenu with the **Destination** of either “physical” (AB device) or “application” (CAPI/TAPI/Message Box Daemon) and a **Type** “all” or “voice”.

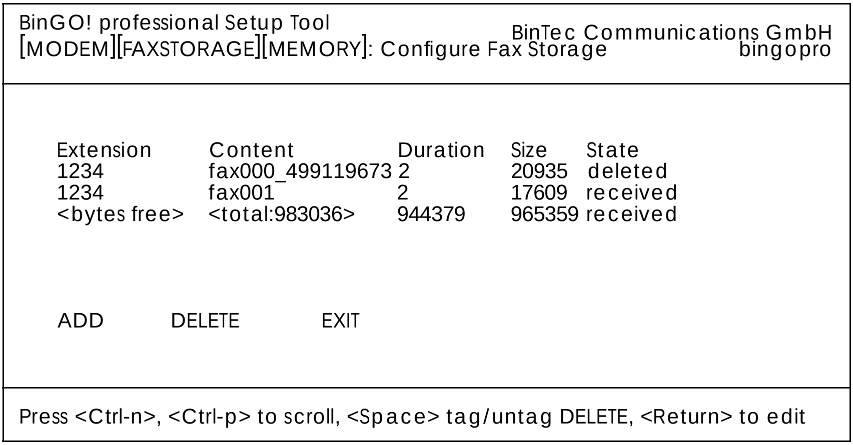
Forward Address = The phone number the faxes shall be forwarded to. When the faxes shall be forwarded to an internal number, e.g. the same via which they were received, the entry must be preceeded by a hash “#”. For example, if you assigned “Prefix # for internal calls” in the PABX Static Settings, you would enter “#1234” . When “Prefix 0 for external calls” is configured you correspondingly have to add a “0” as prefix to an external phone number.
When you do not want that the faxes are forwarded, leave this item empty.

⚠ If forwarding to an external phone number and forwarding is not successfull, there may occur charging problems, because the BinGO! will repeat forwarding the fax until the delivery was successfull.

Poll Mode = Can receive the values ***enable*** and ***disable*** (default).
To allow polling stored fax messages this value must be set to ***enable***. When set to ***disable*** faxpolling is not allowed.

Delete Action = This item can receive the values ***delivered***, ***forwarded*** (default) and ***polled***.
Delivered means that faxes are deleted after they have been delivered (successfully forwarded or polled). When ***forwarded*** is configured, the faxes are deleted after they were forwarded and ***polled*** means that they are deleted after they have been polled. You can always delete a fax manually in Setup Tool´s Memory menu as described below or in the MIB as described under MIB. When faxes are not forwarded or polled and consequently not deleted, they would otherwise use up the capacity of the fax memory.

MODEM → FAX STORAGE → MEMORY →



```
BinGO! professional Setup Tool                      BinTec Communications GmbH
[MODEM][FAXSTORAGE][MEMORY]: Configure Fax Storage                   bingopro


    Extension     Content            Duration   Size    State
    1234          fax000_499119673 2            20935   deleted
    1234          fax001             2          17609   received
    <bytes free>  <total:983036>     944379     965359 received



    ADD          DELETE          EXIT


Press <Ctrl-n>, <Ctrl-p> to scroll, <Space> tag/untag DELETE, <Return> to edit
```

The Memory submenu displays the contents of fax storage. This display is static. Each time you enter this menu the listing is updated. You can also delete single faxes or the complete contents of fax storage.

Extension = The extension, via which the fax was received.

Content = Contains the information ***fax<no>_<remote_faxid>***. Here ***<no>*** is a consecutive number and ***<remote_faxid>*** is the FaxID of the remote fax. When no faxID is available, it is tried to find out the calling number via CLID (Calling Line Identification). When this is also not possible ***<remote_faxid>*** stays empty.

Duration = Contains the number of pages of the fax. Negative entries identifies a fax that was received but had transmission errors (A “-2” means that two pages were correctly transmitted, but the fax message is not complete). A 0 (zero) means that no pages were correctly received (i.e., transmission failure occurred during page one).

Size = Shows the size of the fax in bytes.

State = Can have the values ***receiving***, ***received***, ***sent*** and ***deleted***.

The last entry in the table with the ***Extension "<bytes free>"*** shows the total available memory for faxes (***Content***), and the currently available memory in bytes (***Size***).

Deleting a FAX

To delete a fax (one entry) from the fax storage you can tag it for deletion using the SPACE key and then select DELETE. The menu entry is then marked as deleted.

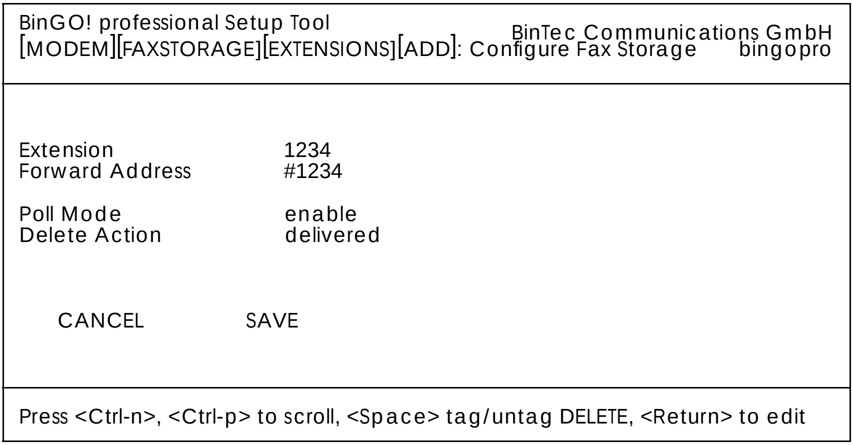
Note that although the entry is initially removed from the list of faxes the actual memory block occupied by the fax is not cleared until the BinGO! requires additional memory space. If you exit and re-enter the Memory submenu the fax may still be listed but will be marked as deleted in the State column.

Deleting all FAXes

To delete the complete contents of fax memory you can tag the last entry (<bytes free>) using the SPACE key and then selecting DELETE. All stored faxes are deleted. In contrast to deleting a single fax as noted above, this will force the BinGO! to immediately delete all faxes (the complete memory area in flash used for FAX messages is cleared).

Configuration Examples

In the following a configuration example for a standard usage of the fax storage, where faxpolling is allowed and faxes are deleted after they have been delivered by forwarding or polling:



```
BinGO! professional Setup Tool                         BinTec Communications GmbH
[MODEM][FAXSTORAGE][EXTENSIONS][ADD]: Configure Fax Storage        bingopro


Extension                 1234
Forward Address           #1234

Poll Mode                 enable
Delete Action             delivered


    CANCEL               SAVE


Press <Ctrl-n>, <Ctrl-p> to scroll, <Space> tag/untag DELETE, <Return> to edit
```

To forward the faxes to a defined phone number and delete them after they have been forwarded and at the same time not allow faxpolling, you must configure the items as following:

Forward Address = is defined
(This item corresponds to ***msgForwardAddr*** variable.)

Poll Mode = disable
(This item corresponds to ***msgForwardPoll*** variable.)

Delete Action = delivered
(This item corresponds to ***msgForwardDeleteAction*** variable.)

You do not forward the faxes and only allow faxpolling. The faxes are deleted after they have been polled:

Forward Address = is not defined

Poll Mode = enable

Delete Action = delivered

The faxes are forwarded and can be polled:

Forward Address = is defined

Poll Mode = enable

Depending on whether the faxes shall be deleted after they have been forwarded, polled or one of both, the item **Delete Action** must be set to **forwarded, polled** or **delivered**

MIB

msgForwardTable

The ***msgForwardTable*** belongs to the ISDN group within the MIB.

```
inx     Extension(*rw)      Addr(rw)            DeleteAction(rw)   Delay(rw)
        Poll(-rw)           PollPassword(rw)

bingopro:msgForwardTable>
```

The ***msgForwardTable*** contains information about the behavior of the fax storage. You can configure which extensions are used to accept fax calls and how these faxes should be forwarded to the receiver.

- ***msgForwardExtension***
  Contains the extension that is used to accept fax calls.

- ***msgForwardAddr***
  Contains the phone number the faxes shall be forwarded to. When the faxes shall be forwarded to an internal phone number, e.g. the extension via which they were received, the entry must be preceeded by a hash “#”, e.g. “#1234”, when the variable ***pabxLocalPrefix*** (***pabxTable***) has the value “#”. When ***pabxLocalPrefix*** is empty and with the variable ***pabxTrunkPrefixPrefix*** (***pabxTrunkPrefixTable***) a escape character (in general “0”) is assigned, an external phone number the faxes shall be forwarded to must be preceeded by this escape character.
  When you do not want the faxes to be forwarded, leave this variable empty.
  If forwarding to an external phone number and forwarding is not successfull, there may occur charging problems, because the BinGO! will repeat forwarding the fax until the delivery was successfull.

- ***msgForwardDeleteAction***
  This variable can receive the values ***delivered***, ***forwarded*** (default) and ***polled***.
  Delivered means that faxes are deleted after they have been delivered (sucessfully forwarded or polled). When ***forwarded*** is configured, the faxes are deleted after they were forwarded and ***polled*** means that they are deleted after they have been polled. Selecting ***forwarded*** or ***polled*** may have the consequence that you have to delete faxes manually by setting the respective ***msgDirState*** variable in the ***msgDirTable*** to ***delete***. When faxes are not forwarded or polled and consequently not deleted, they would otherwise use up the capacity of the fax memory.

- ***msgForwardDelay***
  The time in seconds to wait before the message box tries or retries to send a stored fax to the msgForwardAddr. The default value for this variable is 300 s.

- ***msgForwardPoll***
  Can receive the values ***enable***, ***disable*** (default) and ***delete***. To allow polling stored fax messages this value must be set to ***enable***. When set to ***disable*** fax polling is not allowed. When set to ***delete***, the table entry will be deleted.

- ***msgForwardPollPassword***
  The password you have to use if you like to poll a stored fax. This variable is ignored at the moment.

msgDirTable

The ***msgDirTable*** belongs to the ISDN group within the MIB.

```
inx     Extension(*ro)  Type(ro)    Content(ro)          Time(ro)
        Duration(ro)    Size(ro)    State(-rw)

00      "1234"           fax        "fax001_4991199"     08/25/98 14:50:53
        2               20935       deleted

0        "1234"         fax         "fax002_4991199"     08/25/98 14:52:56
        2               17609       received

0        "<bytes free>" none        "<total:983036>"    08/25/98 14:52:56
        944379          965359      received

bingopro:msgDirTable>
```

The ***msgDirTable*** contains information about the currently stored faxes in the Flash ROM.

- ***msgDirExtension*** (read only)
  Contains the extension, via which the fax was received.
- ***msgDirType*** (read only)
  Can have the values ***fax*** or ***none***, where ***none*** is only displayed, when the document type is not yet recognized.
- ***msgDirContent*** (read only)
  This variable contains the information ***fax<no>_<remote_faxid>***. Here ***<no>*** is a consecutive number and ***<remote_faxid>*** is the FaxID of the remote fax. When no faxID is available, it is tried to find out the calling number via CLID (Calling Line Identification). When this is also not possible ***<remote_faxid>*** stays empty.
- ***msgDirTime***(read only)
  Contains date and time of the received fax in the format:
  mm / dd / yy hh:mm:ss.
- ***msgDirDuration***(read only)
  Contains the number of pages of the fax. Negative values identify a fax that was received but had transmission errors (A “-2” means that two pages were correctly transmitted, but the fax message is not complete). A 0 (zero) means that no pages were correctly received (i.e., transmission failure occurred during page one).
- ***msgDirSize***(read only)
  Shows the size of the fax in bytes.
- ***msgDirState***
  Can have the values ***receiving***, ***received***, ***sent***, ***deleted*** and ***delete***. The first four values are set by the BRICK; the value ***delete*** can be defined by the user manually, when a fax shall be deleted.

The last entry in the table with the ***Extension*** value ***”<bytes free>”*** contains the total available memory for faxes (***Content***), and the currently available memory in bytes (***Size***). You can delete the complete contents of the fax storage by setting the variable ***msgDirState*** of this last entry to ***delete***.

What Was New in Release 4.8.6

Release 4.8 Revision 6

Released: 07.08.98

Features: | Bugfixes: | Detailed Description:

Features

New BRICKware for BinGO! Plus

☞ There is a new separate *BRICKware for BinGO! Plus/Professional*, which in its current version 4.8.5 is included on the BinGO! Plus/ Professional Companion CD respectively can be retrieved from BinTec's WWW server at http://www.bintec.de (Link: FTP-Server).

Priority Voice and Fax Memory

The features Priority Voice and Fax Memory (the later concerns BinGO! Professional only) are not yet implemented in Software Release 4.8.6 and will be included in a future release.

Therefore you should always have a look on our FTP server at http://www.bintec.de for the current software release and documentation.

Microsoft Callback Extension to Mode 3

The Microsoft Callback Control Protocol (CBCP) knows different modes to decide which number is used for callback. This protocol is activated, when there is a call from a Windows95/ NT client.

Up to now Mode 2 was implemented. In Mode 2 (callback to a user-specifiable number) the user is asked, when calling from a Windows95/ NT client, to enter the callback number. This number is then used for callback.

From this release on the MS-CBCP was extended to Mode 3. Mode 3 uses a predefined number for callback.

Which mode is used (Mode 2 or Mode 3) depends on whether there is a predefined number assigned. When there is a predefined number, either a entry in the ***biboPPPDialTable*** for this partner (***Direction***: ***both*** or ***outgoing***; ***Type***: ***isdn*** or ***isdn_spv***) or, when authentication is made via RADIUS and the RADIUS attribute Callback-Number is assigned, then Mode 3 is used. When calling from a Windows95/ NT client the caller is asked in a dialog box to confirm the mode (Mode 3) respectively the callback number. With no number assigned callback is made using Mode 2.

Such it is ensured that a callback is either made using the user-specified or the predefined number.

The variable ***CallBack*** in the ***biboPPPTable*** can be set to ***ppp_offered*** or ***enabled***. But you must notice that with the value set to ***enabled*** no authentication is made during callback.

Also see Microsoft Callback via RADIUS below.

Microsoft Callback via RADIUS

With this new release it is possible to use Microsoft Callback via RADIUS for calls from a Windows95/NT client.

The RADIUS server must be configured as follows:

Service-Type = Callback-Framed

Specifying only the Service-Type means using Mode 2 of the CBCP (user-specifiable number). This configuration assigns the value ***enabled*** to the variable ***biboPPPCallBack*** in the ***biboPPPTable***.

To use Mode 3 (predefined number), that means using a fixed callback number, you must additionally assign a callback number as in the following example:

Service-Type = Callback-Framed

Callback-Number = “392”

The feature Microsoft Callback via RADIUS is only available for an inband identification of the caller (no calling line identification). The same it’s not possible to set the value ***ppp_offered*** for the variable ***biboPPPCallBack***. For the time of the PPP connection there exists a temporary entry in the ***biboPPPTable*** with variable ***CallBack*** assigned the value ***enabled***. This means that there is no additional authentication during callback. In Mode 3 a temporary entry in the ***biboPPPDialTable*** using the defined calling number is generated, too.

IPX RADIUS Extensions

The BRICK now supports dial-up IPX client connections via RADIUS. For a detailed description of this new feature see IPX RADIUS Extensions on page 43.

Frame Relay

Beginning with Release 4.8 Revision 6 Frame Relay is officially supported on the BinGO! Plus / Professional. The BinGO! Plus / Professional can be used as a Frame Relay Switch or a Frame Relay Router and supports the following official and defacto standards:

RFC 1490 *Multiprotocol Interconnect over Frame Relay*
RFC 1293 *Inverse Address Resolution Protocol*
ITU-T Q933a, Appendix II, X6 *Line Management Extensions*
FRF 1.1 *Congestion Management*

Frame Relay requires a separate license to be installed on the BRICK and may be purchased directly from BinTec Communications or your local distributor.

A detailed description of Frame Relay on the BRICK (Setup Tool menus and SNMP shell MIB objects / tables) as well as background information for users not familiar with Frame Relay will in future be contained in the Extended Features Reference, which is available in PDF format via BinTec's FTP server at http://www.bintec.de. Meanwhile you can find a documentation of Frame Relay in the Reference Manual section of the BinGO! Plus / Professional Page on BinTec's FTP server.

X.25

X.25 Window/ Packet Size Negotiation

Now you can decide for each X.25 link, whether a window/ packet size negotiation is made.

x25LkPrNegotiation is the new parameter in the ***x25LinkPresetTable***, which handles this feature. This parameter can be assigned three possible values:

never No negotiation. When a call arrives that does not correspond to the default size, the call is cleared.

always Negotiations are always made.

when_necessary There are only negotiations, when the requested values differ from the default values.

Window/ packet size negotiation settings can also be configured via Setup Tool, see Configuring X.25 Parameters in Setup Tool on page 28.

Configuring X.25 Parameters in Setup Tool

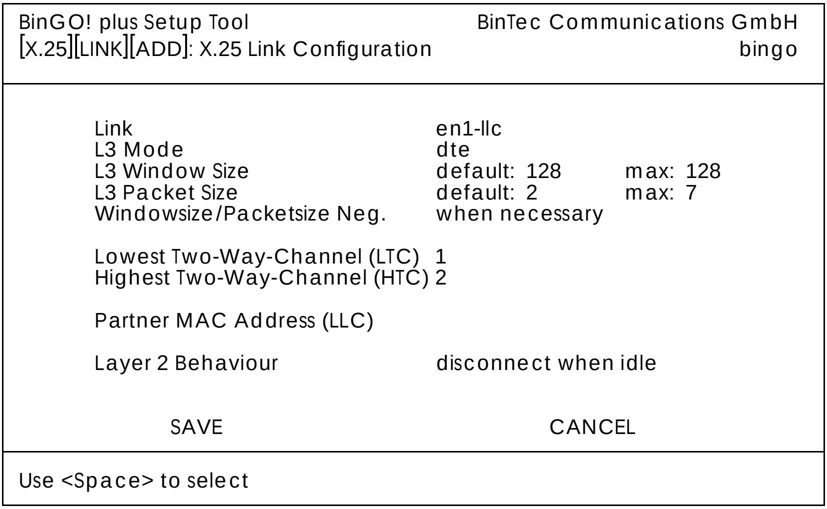
Now it is possible to configure additional X.25 parameters using Setup Tool.

•X.25 Link Configuration

X.25 → Link Configuration → Edit

Here window/ packet size negotiation can be adjusted for an X.25 link. **Windowsize/ Packetsize Neg.** corresponds to the parameter ***x25LkPrNegotiation*** in the ***x25LinkPresetTable***.

Windowsize/ Packetsize Neg. = Decides whether window/ packetsize negotiation is made for this X.25 link. The possible values are **never**, **always** and **when necessary**, where **when necessary** is the default value. The value ***never*** means no negotiation. When a call arrives that does not correspond to the default size, the call is cleared. ***Always*** means negotiations are always made and when ***when necessary*** is



```
BinGO! plus Setup Tool                          BinTec Communications GmbH
[X.25][LINK][ADD]: X.25 Link Configuration                           bingo

      Link                              en1-llc
      L3 Mode                           dte
      L3 Window Size                    default: 128       max: 128
      L3 Packet Size                    default: 2         max: 7
      Windowsize/Packetsize Neg.        when necessary

      Lowest Two-Way-Channel (LTC)  1
      Highest Two-Way-Channel (HTC) 2

      Partner MAC Address (LLC)

      Layer 2 Behaviour                 disconnect when idle


           SAVE                                 CANCEL

Use <Space> to select
```

selected, there are only negotiations, when the requested values differ from the default values.

•WAN Partner

WAN PARTNER → ADD → ADVANCED SETTINGS

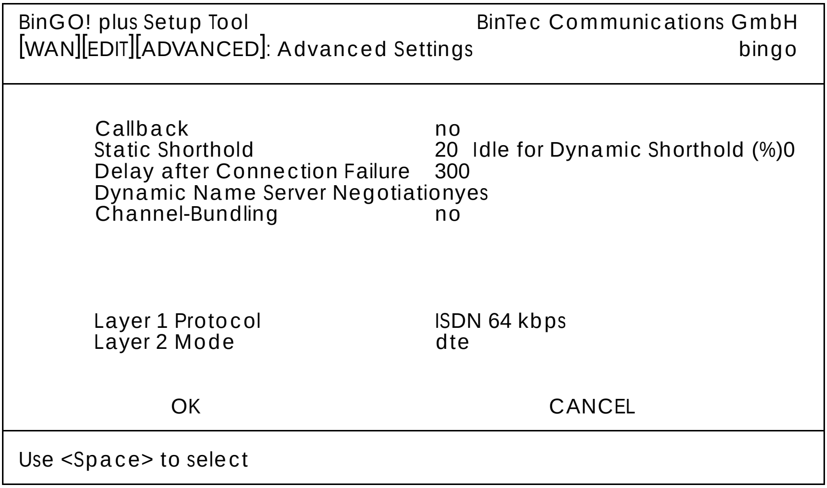
For WAN partners using the protocols X.25, X.25ppp, X.31 B-Channel or X.25 no signalling the Layer 2 Mode can be configured in the advanced settings. The item Layer 2 Mode corresponds to the parameter ***biboPPPLayer2Mode*** in the ***biboPPPTable***.

Layer 2 Mode = Layer 2 Mode can receive the values **auto**, **dte** or **dce**, where **auto** is the default value

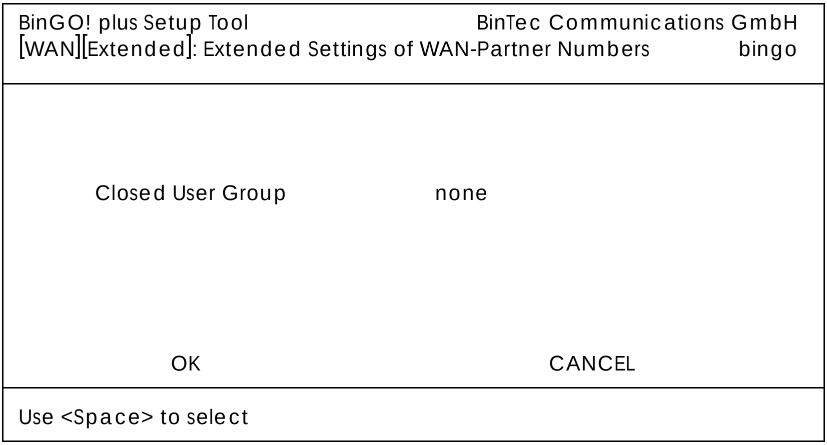
•WAN Partner Numbers Advanced

WAN PARTNER → ADD → WAN NUMBERS →
→ ADD → ADVANCED SETTINGS →

Here the item Closed User Group can be configured. The item corresponds to the parameter ***biboDialClosedUserGroup*** in the ***biboDialTable.***



```
BinGO! plus Setup Tool                              BinTec Communications GmbH
[WAN][EDIT][ADVANCED]: Advanced Settings                                 bingo

      Callback                          no
      Static Shorthold                  20  Idle for Dynamic Shorthold (%)0
      Delay after Connection Failure    300
      Dynamic Name Server Negotiationyes
      Channel-Bundling                  no



      Layer 1 Protocol                  ISDN 64 kbps
      Layer 2 Mode                      dte

            OK                                   CANCEL

Use <Space> to select
```



```
BinGO! plus Setup Tool                              BinTec Communications GmbH
[WAN][Extended]: Extended Settings of WAN-Partner Numbers                bingo



      Closed User Group                 none




            OK                                   CANCEL

Use <Space> to select
```

Closed User Group = The item Closed User Group can be assigned the values **none** or an integer from **1 to 9999**. **None** is the default value.

Active Layer 2 Set Up for Incoming X.25 Calls

Prior to release 4.8.6 the BRICK remained passive during setup of incoming X.25 dialup connections and waited for a SABM (Set Asynchronous Balance Mode) from the caller. Some X.25

implementations however were also waiting for a SABM from the BRICK.

Now the BRICK only waits one second for an incoming SABM. If no SABM is received within this time, the BRICK will send a SABM.

Because of the wait time the probability of a layer 2 setup collision is very small. Standard end-devices handle such collision correctly.

TP0 Bridge Extensions

To make possible connections between TCP clients and an X.25 network the TP0 Bridge feature (RFC 1086 / RFC 1006) is implemented on the BRICK.

With this release two extensions concerning the transmission of X.25 data (RFC 1006) for incoming X.25 connections to the TP0 Bridge have been made.

Firstly with release 4.8.6 NSAP addresses, which are subaddresses of X.25 addresses, can be proofed for incoming X.25 calls. If a listener transfers a NSAP address in the facility field of the listening address, only X.25 calls with the same NSAP address are signaled to the listener.

The second extension concerns the X.25 call indication packet, which is sent as the first packet, when an incoming X.25 connection is established. Now with release 4.8.6 there is a possibility that the listening application gets some information about the contents of the X.25 call indication packet.

To get this data the value of the function byte, (the first byte, the listener sends to the TP0 brigde, see RFC 1086) has to be 66 instead of 2. Then the first data packet, the listener receives on its new established TCP stream has the following format: It consists of 4 Byte TP0 header and the data in the extended X.25 Address Format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| vrsn = 255    |  reserved = 3 |          packet length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        address  type  = 4     | X.25 addr len | X.25 address  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      ...      |      ...      |      ...      |      ...      |
|      ...      |      ...      |      ...      |      ...      |
|      ...      |      ...      |      ...      |      ...      |
|      ...      |      ...      |      ...      |Protocol ID len|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Protocol ID  |      ...      |      ...      |      ...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CUDF length   | Call User Data field ...      |      ...      |
|      ...      |      ...      |      ...      |      ...      |
|      ...      |      ...      |      ...      |      ...      |
|      ...      |      ...      |      ...      |      ...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Facility Length|  X.25 Facilities     ...      |      ...      |
|      ...      |      ...      |      ...      |      ...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IP Filter for TCP State and ICMP Type

The filters for IP access have been enhanced.

ICMP Type

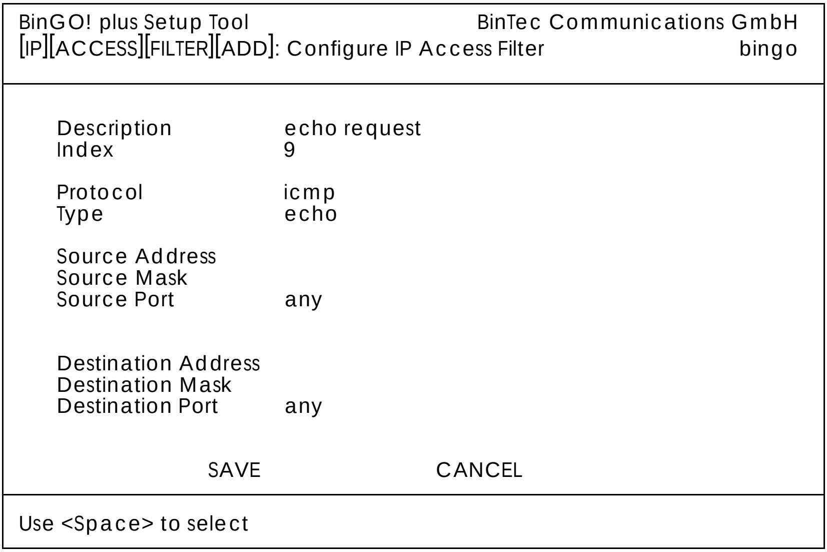
The filters can now be used to filter IP packets in dependence of the ICMP type.

In the ***ipFiltertable*** there is the new variable ***icmptype***, which can be assigned the following values :

echoRep, ***destUnreach***, ***srcQuench***, ***redirect***, ***echo***, ***timeExcds***, ***parmProb***, ***timestamp***, ***timestampRep***, ***addrMask***, ***addrMaskRep***, ***dont_verify*** .

Setup Tool’s Filters menu has also been changed. You can now define filters according to appropriate ICMP types using the Type field after setting the protocol field to “ICMP”.

IP → ACCESS LISTS → FILTERS → ADD



```
BinGO! plus Setup Tool                           BinTec Communications GmbH
[IP][ACCESS][FILTER][ADD]: Configure IP Access Filter                 bingo

    Description              echo request
    Index                    9

    Protocol                 icmp
    Type                     echo

    Source Address
    Source Mask
    Source Port              any


    Destination Address
    Destination Mask
    Destination Port         any

                 SAVE                    CANCEL

Use <Space> to select
```

TCP Connection State

Filters can now be defined based on the state of an TCP Connection.

In the ***ipFilterTable*** there is the new variable ***TcpConnState***, which can be assigned the following values:

dont_verify, ***established***.

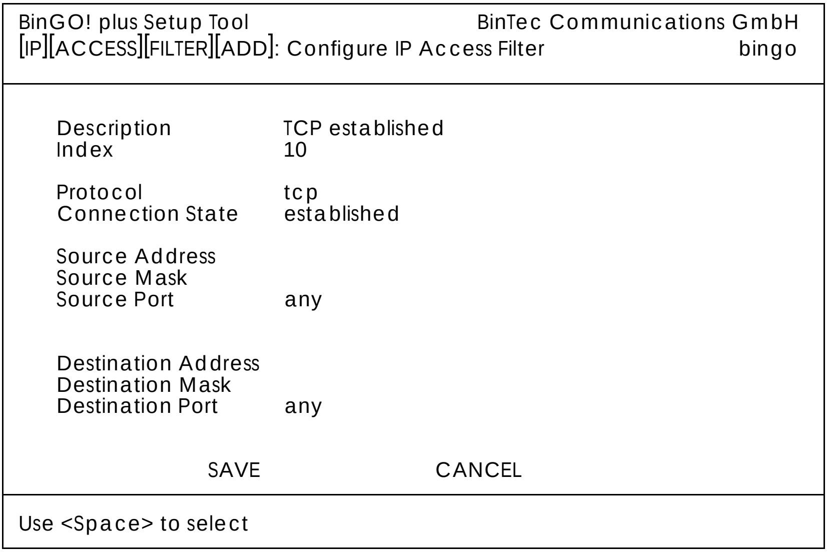
When this variable is set to ***established***, this filter matches for TCP packets,which do not initiate a connection.

A typical application for this filter is to let packets pass through, which belong to connections that were initiated from inside, but discard all other TCP packets. This can be configured by the following rules:

1. rule:	ALLOW (TCP / established)
2. rule:	DENY (TCP / dont_verify)

The configuration in Setup Tool:

IP → ACCESS LISTS → FILTERS → ADD



```
BinGO! plus Setup Tool                              BinTec Communications GmbH
[IP][ACCESS][FILTER][ADD]: Configure IP Access Filter                    bingo

    Description              TCP established
    Index                    10

    Protocol                 tcp
    Connection State         established

    Source Address
    Source Mask
    Source Port              any


    Destination Address
    Destination Mask
    Destination Port         any

                SAVE                      CANCEL

Use <Space> to select
```

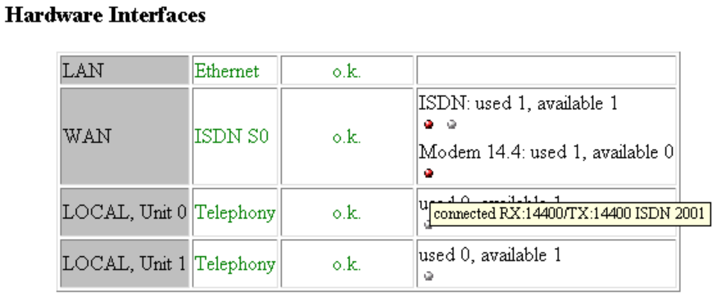
New Trace Command Feature

The trace command has been enhanced. It is now possible to decode HOLD and RETRIEVE messages in the D-Channel.

Status Display for Modems

This feature concerns the BinGO! Professional only.

The web based status page for your BRICK now additionally displays information on installed modems.



Hardware Interfaces

LAN	Ethernet	o.k.	
WAN	ISDN S0	o.k.	ISDN: used 1, available 1 ● ○ Modem 14.4: used 1, available 0 ●
LOCAL, Unit 0	Telephony	o.k.	used 0, available 1 ○ connected RX:14400/TX:14400 ISDN 2001
LOCAL, Unit 1	Telephony	o.k.	used 0, available 1 ○

Under Hardware Interfaces you will find the modems with the respective slot they are installed in. The modem type and modem status are displayed. Each modem used at the moment is marked red and when you move the mouse pointer over the red channel symbol, the rate for receiving and transmitting data in bps is displayed. Additionally you are informed about the ISDN channel used. The four digits xyzz stand for the slot (x), the unit (y) and the ISDN channel used (zz). For the example below this would consequently mean: slot 2, unit 0 and channel 1.

Wildcards for Dialing Numbers

Similar to wildcards for the calling party's address in the ***biboDialTable*** for incoming calls, the variable ***Number*** now can also contain wildcards for outgoing calls. The wildcards for outgoing and incoming calls are defined as follows:

Wildcard	Example	Outgoing Calls	Incoming Calls
*	1234*	is ignored, e.g 1234	matches zero or any string, e.g 1234 or 123467
?	1234?	is replaced by 0, e.g. 12340	matches any single digit, e.g. 12349, 12347
[a-b]	123[5-9]	first digit in the range, e.g. 1235	denotes the range of possible digits to match, e.g. 1235, 1236
[^a-b]	123[^0-5]	range of digits not allowed, first possible digit inserted, e.g. 1236	denotes the range of excluded digits to match, e.g. 1236, 1237
{ab}	{00}1234	inserted for outgoing calls, e.g. 001234	optional string to match, e.g. 001234, 1234

The advantage is, that now you can use one entry for the variable ***Number*** for incoming and outgoing calls. For example {0}91196790 will generate 091196790 for outgoing calls and will accept 091196790 and 91196790 for incoming calls as valid CLID.

Link Quality Monitoring

By the help of Link Quality Monitoring (LQM defined in RFC 1989) it is possible to exchange information within a PPP connection to draw conclusions about the underlying connection quality.

This information is typically transmitted periodically to the partner as so-called Link Quality Reports (LQR). The interval (Reporting Period) is agreed upon during the LCP negotiation.

Link Quality Monitoring can be useful to examine e.g. modem connections. (With unreliable modem connections it can happen that because of CRC errors no more data can be transmitted.)

For detailed information on the new feature Link Quality Monitoring see the section Link Quality Monitoring in Detailed Feature Descriptions on page 46.

Extended Syslog Messages

This feature concerns the BinGO! Professional only.

The syslog messages have been extended for a detailed analysis of fax connections.

TAPI Names

In TAPI applications of BRICKware for Windows the TAPI names for the physical terminals (POTS) receive unambiguous names now. In case of two POTS at one BRICK, both having the same MSN, they are named Phone1 and Phone2 in respect to the socket they are connected to.

Signalling Phone Calls

With this software release phone calls to physical devices (POTS) are also signalled to the CAPI application of the same user. This means that now one entry for the physical destination in the PABX/Extensions menu of the Setup Tool is sufficient.

CAPI Applications

Now a direct dial-in to CAPI applications is possible. This Feature is significant for point-to-point connections.

Bugfixes

Known Bug

- Dial-up connections for RADIUS-Users
  Interfaces configured to use `ip_lapb` encapsulation, using the following entry in /etc/raddb/users,
  `BinTec-biboPPPTable = "Encapsulation=ip_lapb"`
  are sometimes rejected by the BRICK.

TP0 Bridge

- Sometimes it happened that it was not possible to establish an initial TP0 Bridge connection (RFC 1086) between the TCP client and the BRICK.

  This bug has been fixed. In this context the TP0 Bridge syslog messages have been changed, too.

X.31 in D-Channel

- When CAPI applications were using X.31 in D-Channel for X.25 packet switching, the variable ***AssignedTo*** in the ***isdnDChanX31Table*** was not considered (Setup Tool: Advanced settings for the WAN interface).

  This problem has been fixed.

  The variable ***AssignedTo*** can be assigned the following values:

packet_switch	The TEI may be used only by the X.25 router.
capi	The TEI may only be used by a CAPI application and this TEI has to be configured also in the CAPI application.
capi_default	The TEI may be used only by a CAPI application and the BRICK overwrites the TEI value that is configured in the application.
delete	Sets the table entry to delete.

CAPI

- If a CAPI application used an X.25 protocol, it wasn't any longer informed about incoming X.25 calls after it has sent a CONNECT_B3_REQ message to establish an outgoing X.25 connection. In this case the CONNECT_B3_Ind message, the application has to receive got lost.

  This bug has been fixed.

STAC Compression on Multilink PPP Interfaces

- According to RFC 1974 (*PPP STAC LZS Compression Protocol*) there are several check modes to keep the compressor and decompressor histories in synchronisation even in the absence of a reliable link to guarantee the sequential transmission of data.

  In rare cases it still happened that, when ***biboPPPCompression*** = stac (RFC 1974, check mode 3) was used on MultiLink PPP interfaces the history re-synchronisation process sometimes came to a state where decompression histories were out-of-sync and user data could no longer be transmitted over the line.

  This problem has been fixed in the current release.

Spaces in biboPPPLoginString

- There appeared problems in the login procedure configuration (especially for Compuserve users), when strings like passwords or login names were containing spaces. That was because spaces are used as internal flags to handle the login procedure.

  To handle this problem blank spaces in strings, which are part of the variable ***LoginString*** in the ***biboPPPTable***, must be preceded by a backslash as shown in the following example for the string "pass word":

```
inx LoginString(rw)
00 "-d1 \n e: CIS\n D: name/go:pppconnect\n wor -d1 pass\ word\n PPP"
```

This must be considered, when the variable ***LoginString*** is configured via SNMP. In the Setup Tool no additional backslashes have to be entered, when configuring the items Host, User ID and Password in the menu [WAN][EDIT][ADVANCED][PROVIDER].

Setting Administration Status to Down

- When by “ifconfig down” or via the Setup Tool the variable ***ifAdminStatus*** was set to down for an active interface, the variables in the PPP accounting syslog message containing PPP connection information only had the value 0.

  This bug has been fixed. Now the syslog message contains the correct values.

Setup Tool

- When a PPP interface was resetted in the interface monitor in [MONITOR][INTERFACES][EXTENDED], there could occur a reboot of the BRICK with large configurations.

  This bug has been fixed.

- When a default route was configured for a WAN partner interface in the [IP][ROUTING] menu or the SNMP shell and afterwards the [WAN][EDIT][IP] menu was opened again for this WAN Partner and left with SAVE, this default route was deleted.

  This bug has been fixed.

L1 and L2 LEDs

- The LEDs L1 and L2 of the BinGO! Plus / Professional did not switch to the on state, when external phone calls were performed.

  This bug has been fixed.

Disconnecting an ISDN Call

- An ISDN call could not be interrupted by setting the variable ***isdnCallState*** in the ***isdnCallTable*** to ***null*** or choosing **release** in the ISDN Monitor (Monitoring and Debugging / ISDN Monitor) of the Setup Tool.

This bug has been fixed.

Outgoing Call

- When during autoconfiguration (protocol detection) an outgoing call (data or voice) was started, a reboot of the BRICK occured.

  This bug has been fixed.

OSPF Licenses

- In the last Software Release it was not possible to enter OSPF licenses, i.e. OSPF was not activated, after the license had been entered.

  This bug has been fixed.

Trunk Port Configuration

- When changing the configuration of the trunk port (in the pabxTrunkTable or the WAN Interface menu of the Setup Tool), there could occur a reboot of the BinGO! Plus / Professional.

  This bug has been fixed.

Dynamic Shorthold

- Information on ISDN charges was not transmitted to the subsystems (PPP). That was why the Dynamic Shorthold feature was not active.

  This problem has been fixed.

- When combining dynamic B-Channel Bundling with optimally making use of the charging intervals (by dynamic shorthold), there was the problem that, when reducing the bandwith, the current charging intervals were not taken into consideration.

  This problem has been fixed.

To optimize charges now the bandwith is reduced by disconnecting a B-Channel only short before a new charging interval.

Detailed Feature Descriptions

IPX RADIUS Extensions

The BRICK now supports dial-up IPX client connections via RADIUS. Support for IPX links via RADIUS has been tested using Merit's AAA RADIUS server 3.5.6. The examples shown below can be used with the Merit server, for use with other servers consult your local documentation.

Assuming a RADIUS server has been configured in Setup Tool's IP ➝ RADIUS SERVER menu (or the ***radiusServerTable*** from the SNMP shell), and the RADIUS server can successfully authenticate the caller, dial-up links can be setup to support IPX networking using standard IPX or BinTec-specific RADIUS attributes mentioned below.

Standard Attributes

Framed-IPX-Network

This attribute defines a transfer network for the IPX link. Setting this attribute to "Framed-IPX-Network = 8" effectively sets the IPX network number for the transfer network to "0:0:0:8".

Normally, when this attribute is set to "fffffffe" the calling host is assigned a network number from an existing address pool, currently this feature is not supported on the BRICK. To disable a transfer network for the IPX WAN link you can set this attribute to "0" (no transfer network, unnumbered RIP).

RIP / SAP updates: If the "Framed-IPX-Network" attribute is used entries in the BRICK's ***ripCircTable*** and ***sapCircTable*** are configured using default settings for RIP / SAP updates (triggered+piggybacked).

BinTec-specific Attributes

BinTec-specific RADIUS attributes added in release 4.8.3 are defined below. For additional information, see also the section *Using BinTec-specific Attributes* on page 45.

BinTec-ipxCircTable	Creates or modifies ***ipxCircTable*** entries.

BinTec-ripCircTable	Creates or modifies ***ripCircTable*** entries
BinTec-sapCircTable	Creates ore modifies ***sapCircTable*** entries.

Example IPX over RADIUS /etc/raddb/client Entries

The definitions below could be used for a dial-up IPX link between two BRICKs. RIP / SAP updates will be performed via the default triggered+piggybacked mode.

```
kornburgxl
    Password = “access2roth”, Framed-IPX-Network = 0
    Framed-Protocol = PPP, Idle-Timeout = 300,
rothxl
    Password = “access2kornburg”, Framed-IPX-Network = 0
    Framed-Protocol = PPP. Idle-Timeout = 300,
```

If the router rothxl supports IPX WAN links but requires a transfer network (0:0:0:9) this definition could be used instead.

```
rothxl
    Password = “access2kornburg”, Framed-IPX-Network = 9
    Framed-Protocol = PPP. Idle-Timeout = 300,
```

If the required IPX services are statically configured, RIP and SAP can be disabled for the WAN link using the BinTec-specific options shown below.

```
kornburgxl
    Password = “access2roth”, Framed-IPX-Network = 0
    Framed-Protocol = PPP, Idle-Timeout = 300,
    BinTec-ripCircTable = “ripcircstate=off”,
    BinTec-sapCircTable = “sapcircstate=off”
```

The definition below could be used for a dial-in Windows Client that does not support IPX WAN links. In this example, setting “Update = 0” disables periodic updates for active links.

```
winlaptop
    Password = “486pentium?”
    Framed-Protocol = PPP, Idle-Timeout = 300,
    Bintec-ipxCircTable = “netnumber=0:0:0:a ipxcirctype=ipxcpWS”
    BinTec-ripCircTable = “Update=0 AgeMultiplier=10000”,
    BinTec-sapCircTable = “Update=0 AgeMultiplier=10000”
```

RIP/SAP Updates for RADIUS interfaces

Since RADIUS interfaces are only available as long as the respective client is connected please note the following effects this may have on links configured for active RIP and SAP updates.

- Access to services on the dial-in client's LAN (from hosts on the RADIUS client's LAN) may not be reliable.
- IPX clients cannot be informed of changes (routing or service advertisements) unless they are actually connected when the change occurs.

This may lead to a state where a server appears as being present on the remote network but is no longer available. The preferred solution, albeit time-consuming, to this problem is to statically configure the required routes and services on the clients and to disable RIP/SAP updates.

Using BinTec-specific Attributes

Each of the BinTec-specific RADIUS attributes corresponds to a MIB table. Supported BinTec-specific attributes can be used in your server's /etc/raddb/users file. The attribute definitions must also be added to your dictionary file (normally found in /etc/raddb). To modify a MIB table entry you must use the following syntax:

<*BinTec-Option*> = "*variable*1=*value*1 … *variable*n=*value*n"

An example authentication line from a RADIUS /etc/raddb/users file might look like this:

```
Service-Type = Framed,
BinTec-biboPPPTable = "DynShorthold=50 IpAddress=static",
BinTec-ipNatPresetTable = "Protocol=tcp extport=1050 intport=100"
```

Also, when using these attributes please note:

- The table entry's *ifIndex* is set automatically and can't be influenced.
- The entries are not case-sensitive.
- You must not use blank spaces before or after »=« signs inside the double quotes.
- Attributes support either **static** or **dynamic** mode.

Static mode modifies existing table entries while dynamic mode creates a new table entry. All variables you want to create (dynamic) must be defined in one line.

Link Quality Monitoring

Because Link Quality Monitoring as described under Features is specified within LCP negotiation, i.e. before the authentication of the partner, for the configuration of incoming calls a distinction must be made between inband and outband identification.

In case of outband identification (CLID/ outband RADIUS) and for outgoing calls the LQM is activated by setting the variable ***biboPPPLQMonitoring*** in the ***biboPPPTable*** to on. When a RADIUS server is used the variable is set by the help of the BinTec dictionary.

For incoming calls identified inband (identification by the internal ***biboPPPTable*** or via RADIUS server) the variable ***biboPPPProfileLQMonitoring*** in the ***biboPPPProfileTable*** must be set to on.

After a successful LCP negotiation for every link of a temporary connection additionally to the entry in the ***biboPPPLinkTable*** a correlating entry in the ***biboPPPLQMTable*** is generated. Both entries can be uniquely assigned to each other by the ***IFIndex*** respectively the ***CallReference*** value.

The ***biboPPPLQMTable*** is a new table and is described in detail in the following.

biboPPPLQMTable:

```
inx IfIndex(*ro            CallReference(ro)      ReportingPeriod(ro)
    OutLQRs(ro)            OutPackets(ro)         OutOctets(ro)
    InLQRs(ro)             InPackets(ro)          InOctets(ro)
    InDiscards(ro)         InErrors(ro            PeerOutLQRs(ro)
    PeerOutPackets(ro)     PeerOutOctets(ro)      PeerInLQRs(ro)
    PeerInPackets(ro)      PeerInOctets(ro        PeerInDiscards(ro)
    PeerInErrors(ro        LossedOutLQRs(ro)      LossedOutPackets(ro)
    LossedOutOctets(ro)    LossedPeerOutLQRs(ro)LossedPeerOutPkts(ro)
    LossedPeerOutOcts(ro)
```

The ***biboPPPLQMTable*** contains statistical information for each current PPP link on the system. Only the system can add or delete entries to this table.

Entries are created by the system each time a new PPP link was established and LQM was negotiated successfully.

Entries are removed by the system, when the corresponding PPP link is disconnected.

For detailed information on the meaning of the single variables see the MIB Reference on the BinTec Website at http://www.bintec.de

What Was New in Release 4.8.5

Release 4.8 Revision 5

Released: 09.06.98

Features: | Bugfixes: | Detailed Description:

Features

New BRICKware for BinGO! Plus

☞ There is a new separate *BRICKware for BinGO! Plus*, which in its current version 4.8.5 is included on the BinGO! Plus/ Professional Companion CD respectively can be retrieved from BinTec's WWW server at http://www.bintec.de (Link: FTP-Server).

New ISDN Supplementary Services

*8# *Call Transfer*

When you have one call on hold (internal or external) by having dialed R or *0# and set up a second call (internal or external), you can connect these two calls by dialling *8#. This function is available as soon as you hear the ringing for the second, active call.

☞ If the second, active call is an external call the availability of this feature depends on your telephone company's supplementary service ECT (Explicit Call Transfer).

Bugfixes

CAPI

- In release 4.8.4 the ***capiConfigTable*** was empty and therefore the configuration parameters for the remote CAPI couldn't be defined. Now there are default entries for all parameters that can be changed.

  This bug has been fixed.

ISDN Configuration

- A system panic sometimes occurred, when ISDN autoconfiguration was enabled, but the ISDN cable wasn't plugged in.

  This bug has been fixed.

Status Information

- System information for the WAN and POTS interfaces was not displayed correctly in the BinGO!'s status page. Also a system panic sometimes occurred, when the HTTP daemon accessed the ***pabxTable*** or the ***pabxPrefixTable***.

  These bugs have been fixed.

ISDN Accounting

The local and remote numbers were improperly reported in the ISDN accounting messages. This resulted from wrong entries for these two items in the ***isdnCallTable*** and the ***isdnCallHistoryTable***.

This bug has been fixed.

What Was New in Release 4.8.4

Release 4.8 Revision 4

New BRICKware for BinGO! Plus

☞ Beginning in Release 4.8 Revision 4 a new version of *BRICKware for BinGO! Plus* has been made available that includes special support for BinGO! Plus systems. The most recent *BRICKware* version can always be retrieved from BinTec's WWW server at http://www.bintec.de (Link: FTP-Server).

New ISDN Supplementary Services

In addition to the supplementary services explained on page 128 of your *BinGO! Plus User's Guide* there are several new functions available. These are marked by a bar on the right hand side of the following description.

∗∗ *dial a single ∗ when necessary*

∗0# *same as the Recall key* (often marked R or Hold)
☞ If your telephone is equipped with a special R key, you can of course use this key instead of dialling ∗0#.

∗1# *disconnect the current call*
Convenient when you have an active call and one call on hold. Dialling ∗1# then terminates the active call and recalls the held call.

∗2# *enable call-waiting*
If you have an active call you will be informed of a waiting call by a call-waiting tone.
After each power-up or reboot of your BinGO! Plus call waiting will be enabled automatically.

∗3# *disable call-waiting*
Additional incoming calls will be refused (*User busy*) when there is an active call. Calls already waiting will also be refused immediately.

∗4# *call forwarding …*

∗4∗0∗‹*No.*›# … *always*
∗4∗1∗‹*No.*›# … *when busy*
∗4∗2∗‹*No.*›# … *when no answer*

where ‹*No.*› is the telephone number to forward the call to.

The codes ∗4∗0∗#, ∗4∗1∗#, and ∗4∗2∗# disable the corresponding call forwarding setting.

☞ You may need to order the *call forwarding* feature from your telephone company before you can use it.

Also note that using *call forwarding* may incur additional charges.

Contact your telephone company for details.

∗5# *three-party conference*

When you have an active call and one call on hold, dialling ∗5# will connect you with both *external* calls in a three-party conference.

☞ You may need to order the *three-party conference* feature from your telephone company before you can use it, and using *three-party conferences* may incur additional charges.

Contact your telephone company for details.

∗6# *terminate three-party conference*

When you have established a three-party conference by dialling ∗5# you can terminate it again with ∗6#. This will return you to the state the calls were in prior to dialling ∗5#, i.e. one external call is on hold, the other external call is connected.

You can then switch between the two calls using the ∗0# combination (Hold).

∗7# *CLIR for the following call*

Prevents your ISDN number from being displayed on your partner’s display for the next call only (CLIR is short for Calling Line Identification Restriction).