

# RELEASE NOTE BIANCA/BRICK-XS and BIANCA/BRICK-XS *office*

September 3, 1997

## New System Software: *Release 4.5 Revision 5*

This document describes the new features, enhancements, bugfixes, and changes to the BIANCA/BRICK-XS and XS office System Software since Release 4.4.8.

<b>New in Rel. 4.5.5</b>	Upgrading System Software. . . . .	2
	ISDN / V.110 . . . . .	3
	(XS office) CAPI - DTMF Tone Recognition . . . . .	4
	(XS office) CAPI - Improved Hardware Fax Support . . . . .	5
	HTTP password / Setup Tool . . . . .	5
	Bugfixes. . . . .	6
<b>Changes in previous Releases</b>	(XS office) CAPI Modem connection support . . . . .	8
	New capiMultiControllerTable . . . . .	8
	(XS office) Hardware fax. . . . .	9
	HTTP / Security . . . . .	9
	ISDN / DDI. . . . .	10
	New x25_ppp_opt encapsulation. . . . .	10
	New field Layer2Mode . . . . .	11
	PPP Modem connections . . . . .	11
	Incoming Call Answering Menu . . . . .	11
	WAN Partner / Outgoing Calls. . . . .	13
	Enhancements for RFC 1006 support . . . . .	14
	Enhancements for RFC 1086 Support . . . . .	15
	Closed User Group support . . . . .	16
	New minipad options . . . . .	17
Bugfixes. . . . .	18	

All user documentation is available in Adobe's PDF format via BinTec's HTTP server at <http://www.bintec.de>.

## Upgrading System Software

1. Retrieve the current system software image from BinTec's HTTP server at <http://www.bintec.de>.
2. With this image you can upgrade the BIANCA/BRICK-XS 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 *BRICK-XS User's Guide* under *Firmware Upgrades*.
3. Once you've installed release 4.5 Revision 5 you may want to retrieve the latest documentation (in Adobe's PDF format) which is also available from BinTec's FTP server 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 HTTP server.

# What's New in Revision 5

## 4.5 Revision 5:

Released: 03.09.97

### *Features:*

#### ISDN / V.110

BIANCA/BRICK-XS internet and BIANCA/BRICK-XS office now fully support asynchronous bit rate adaptation according to V.110.

Asynchronous bit rate adaptation is often used in communication with terminal adapters and for connecting to GSM networks from the ISDN. The BIANCA/BRICK-XS is fully compatible with the V.110 standard.

#### Configuration

V.110 support can be configured individually for each partner in the *biboPPPLayer1Protocol* variable of the ***biboPPPTable*** (in **Setup Tool** this is the *Layer 1 Protocol* setting in the [WAN Partner][EDIT][Advanced Settings] menu).

v110_1200	V.110 bit rate adaptation (asynchronous 1200 baud, 8,N,1)
v110_2400	V.110 bit rate adaptation (asynchronous 2400 baud, 8,N,1)
v110_4800	V.110 bit rate adaptation (asynchronous 4800 baud, 8,N,1)
v110_9600	V.110 bit rate adaptation (asynchronous 9600 baud, 8,N,1)
v110_19200	V.110 bit rate adaptation (asynchronous 19200 baud, 8,N,1)
v110_38400	V.110 bit rate adaptation (asynchronous 38400 baud, 8,N,1)

For outgoing ISDN calls the Layer1Protocol setting is signalled via the D channel.

## Reception of Incoming Calls

For ISDN partners that can be identified by the ISDN Calling Party's Number (i.e. outband), the *Layer1Protocol* settings will be adjusted according to their [*WAN Partner*] / *biboPPPTable* entry (see previous section).

If the caller can not be identified by his calling number, identification must be performed "inband" using PPP. The setting of the *Layer1Protocol* for incoming calls can then be performed in two different ways.

- In most cases you can specify a generic ppp entry for the ISDN number you wish to accept incoming PPP and V.110 calls on (from the SNMP shell this is done in the *isdnDispatchTable*, setting *Item* to **ppp** and *LocalNumber* to the ISDN number for routing; in Setup Tool go to the slot of your ISDN interface, and then in the [*Incoming Call Answering* >] menu add an entry with *Item* set to **PPP (routing)** and *Number* set to the local ISDN number for routing).

The Layer 1 parameters are then automatically taken from the D channel signalling elements of the incoming call.

- In some cases these signalling elements are not transmitted correctly (e.g. through certain switching stations and PABXs). You can then setup a different dispatch entry with a different ISDN number for each V.110 speed you wish to support (as described above). The following values for *Item* are supported:

```
ppp_v110_1200
ppp_v110_2400
ppp_v110_4800
ppp_v110_9600
ppp_v110_19200
ppp_v110_38400
```

## (XS office) CAPI - DTMF Tone Recognition

BIANCA/BRICK-XS office can now recognize the following DTMF tones with the built-in hardware modems:

0-9, *, #, A-D	Touchtones
X	Fax modem calling tone (1100 Hz, T.30)
e	European data modem calling tone (1300 Hz, V.25)
f	Bell data modem answer tone (2225 Hz)
Y	Modem and Fax answer tone (2100 Hz, V.25 / T.30)

### (XS office) CAPI - Improved Hardware Fax Support

BIANCA/BRICK-XS office now also supports fax transmissions at 12,000 bps and 14,400 bps.



The current revision supports Error Correction Mode (ECM) for outgoing calls with speeds of up to 12,000 bps—if you select 14,400 bps and ECM, the BRICK-XS office will automatically transmit at 12,000 bps.

When selecting a fax speed of 9,600 bps or 7,200 bps from a CAPI application BRICK-XS office automatically uses the V.29 modulation, V.17 is only used for 12,000 bps and 14,400 bps.

## ***Enhancement:***

### HTTP password / Setup Tool

The separate HTTP password (see notes to Rel. 4.5.3) can now also be configured from the Setup Tool. It's the *HTTP Server Password* entry in the [System] menu.

## Bugfixes

### CAPI

- With the introduction of the *capiMultiControllerTable* there occurred an error. When a CAPI application requested a connection to CAPI controller  $n$ , the reply contained a PLCI with controller number  $n+1$ , which was not accepted by some applications.  
The PLCI now contains the correct controller number  $n$ .
- The system rebooted, when it received incorrectly coded DATA\_B3\_REQ messages from an RCAPI application, because the message length was not checked.  
This is no longer the case.
- The system also rebooted, when CAPI2\_RESET\_B3\_REQ messages contained an empty NCPI field.  
This bug was fixed.

### PPP

- Encapsulation x25\_ppp: The appropriate *biboPPPLinkTable* entries will now be deleted after closing a dialup connection. These entries were not deleted in Rel. 4.5.3.
- *OperStatus* in the *IfTable*: In releases 4.5.1 to 4.5.3 this state was immediately changed from **blocked** to **dormant**, ignoring the *biboPPPBlockTime*, which resulted in constant connection setup retries.  
The *biboPPPBlockTime* is now handled correctly.
- (Re)loading configuration files: If the *IfTable* contains entries where *AdminStatus=dialup*, this state will now be changed to *AdminStatus=up* when loading the file.  
In previous releases the *AdminStatus* remained **dialup**, effectively blocking outgoing connections on these interfaces after a reload.

### SNMP

- If no IPX license was installed, SNMP used up more and more memory.  
SNMP now correctly releases all memory areas it no longer needs, and additionally makes a »garbage collection« of

the dynamically allocated memory areas from time to time. This is indicated by a special Syslog Message.

#### X.25

- Closed User Groups / minipad: All Closed User Group parameters of the minipad program now also accept values from 100 to 1000.

## Changes in previous Releases

### 4.5 Revision 3:

Released: 01.08.97

### *Features:*

#### CAPI

##### (XS office) CAPI Modem connection support

Both CAPI 1.1 and CAPI 2.0 now support connections using the built-in modem. At the moment the parameters of modem profile 1 from the ***mdmProfileTable*** (also accessible via the [MODEM] menu of the Setup Tool) are used.

Full support of CAPI modem parameters (B2 Configuration for B2 protocol 7, NCPI for B3 protocol 7) will be available in an upcoming release.

##### New *capiMultiControllerTable*

A new table, the ***capiMultiControllerTable***, was added to the CAPI group to enable the use of CAPI with different ISDN controllers at the same time.



As your BRICK has exactly one ISDN controller this table is of interest only for special USA ISDN configurations.

This table contains mappings between controller numbers used by CAPI applications and the ISDN stacks available on the BRICK (i.e., the *Number* field of the ***isdnStkTable***). The *Version* field specifies whether an entry applies to a ***capi11*** or ***capi20*** application.

If no CAPI 1.1 entry is defined, CAPI 1.1 applications are assigned *isdnStkNumber n* where n is the controller number requested by the application.

If no CAPI 2.0 entry is defined, CAPI 2.0 applications are assigned *isdnStkNumber n-1* where n is the controller number requested by the application.



Creating entries: Entries are created by assigning a value to the *capiControllerNumber* object.

Deleting entries: An entry can be removed by assigning the value **delete** to its *capiControllerVersion* object.

The fields of the table have the following meaning:

*Number* The controller number requested by the CAPI application.

*StkMask* This binary number defines the ISDN stack(s) to use for the specified CAPI 1.1 or CAPI 2.0 applications. Each bit corresponds to one entry (stack) in the *isdn-StkTable*, the rightmost bit selects entry 0, the next bit selects entry 1, and so forth. For example, **Number=1 StkMask=0b1101 Version=capi11** means: allow CAPI 1.1 applications requesting ISDN controller 1 to use ISDN stacks 0, 2 and 3.

*Version* Specifies which CAPI applications (version 1.1, or 2.0) this entry applies to.

Set this field to **delete** to delete this entry.

### (XS office) Hardware fax



Please note that due to software limitations the current system software release 4.5.3 supports fax connections of up to 9600 bps only.

### HTTP / Security

The *bintecsec* table now also contains a *biboAdmHttpPassword*, which has to be entered when you want to view system tables from the http status page of your BRICK. The http password defaults to **bintec**.



After entering this password, you can view all system tables apart from the *bintecsec* table from a WWW browser; therefore you should change the password from its default value (this has

to be done from the SNMP shell at the moment), because otherwise anybody knowing the **bintec** password can spy out your system configuration.



Some WWW browsers require that you enter a user name, before accepting the http password.  
If this is the case, please use **http** as a user name.

## ISDN / DDI

Direct Dialling In (DDI) is now supported for point-to-point ISDN accesses (*Anlagenanschluß* in Germany).

This feature is needed for accepting calls to a point-to-point access from the analogue telephone network, e.g. from a modem.

With DDI the BRICK can collect the digits of the called party number until a matching entry in the ***isdnDispatchTable*** is found. When the call is initiated from an analog device the digits usually arrive one by one.



You have to set the *Mode* in the ***isdnDispatchTable*** to **left\_to\_right (2)** to enable DDI.

For an explanation of how to configure this feature from the **Setup Tool** please refer to section *Incoming Call Answering Menu* on page 11.

## PPP

### New x25\_ppp\_opt encapsulation

There is a new encapsulation type, *x25\_ppp\_opt*, which can be used in the ***biboPPPTable***. It provides a special case of the *x25\_ppp* encapsulation.

The *x25\_ppp\_opt* encapsulation enables your BRICK to determine whether an incoming call is an X.25 call or a PPP call even if no outband authentication (by CLID) is possible. This is done by scanning the first incoming data packet.

Dial-in partners which can not be authenticated outband (CLID) will then be given an X.25 connection via ISDN, or optionally a PPP connection, when they can be authenticated in-band by using CHAP or PAP.



Concurrent use of X.25 and PPP encapsulation is not possible.

Note that you will need one WAN partner definition for X.25, where *x25\_ppp\_opt* is selected, and one or more for PPP connections (authentication via PAP, CHAP or RADIUS).

### New field Layer2Mode

The *Layer2Mode* variable *Layer2Mode* was added to the ***bi-boPPTable***.

This object specifies the layer 2 mode to be used for a connection. It is only relevant, if the Encapsulation involves a LAPB protocol, this is the case for *x25*, *x25\_ppp*, *x25\_ppp\_opt*, *ip\_lapb*, *lapb*, *x31\_bchan*, *x75\_ppp*, *x75btx\_ppp*, and *x25\_nosig*.

The default value of this object is *auto*.

For dialup connections, the layer 2 mode will then be DTE on the calling side and DCE on the called side.

For leased lines the layer 2 mode is set at lower layers (for example *isdnChType* in the ***isdnChTable***). When this object is set to *dte* or *dce*, the layer 2 mode will always be DTE or DCE, regardless of the call direction or the settings at the lower layer.

### PPP Modem connections

PPP now also accepts modem connections signalled as »ISDN speech 3.1 kHz«. These capabilities are signalled by some PBXs with analog ports.

## Setup Tool

### Incoming Call Answering Menu

The [*Incoming Call Answering*] menu was improved from the one shown in your printed manuals. Its purpose remains the

same; however, it now contains a list of arbitrary entries instead of a mask with only a few possible variations. The settings from this menu show up in the *isdnDispatchTable*.

The entries in this list are used to distribute incoming ISDN calls received the ISDN to different services. The BRICK distinguishes incoming calls based on the »Called Party's Number« transmitted with each ISDN call.

Select [ADD] from the [CM-1BRI] [Incoming Call Answering] menu to create a new list entry.

BIANCA/BRICK-XS Setup Tool		BinTec Communications GmbH	
[WAN][INCOMING][ADD]: Conf. Incoming Call Answering		mybrick	
Item Number Mode	PPP (routing)		
	right to left		
		SAVE	CANCEL
Use <Space> to select			

**Item** = the ISDN service you want to use for this call. You can select one of the following:

Value	Meaning
PPP (routing)	Default value, good for all PPP connection types listed below (except for the specific PPP Modem Profile 2 ... 8 settings) if the calls are signalled correctly (as is the case in most of Europe). <i>If in doubt, try this value.</i>
ISDN Login	login service

Value	Meaning
PPP 64k	64kbps PPP data connection
PPP 56k	56kbps PPP data connection (not supported by the feature module)
PPP Modem	selects Modem Profile 1 as configured in the [MODEM] menu
PPP DOVB	<u>d</u> ata <u>t</u> ransmission <u>o</u> ver <u>v</u> oice <u>b</u> earer; useful e.g. in the US where voice calls sometimes cost less than data connections
PPP V.110 (1200 - 38400)	bit-rate adaption according to V.110 (1200 bps, 2400 bps, ..., 38400 bps)
Pots	only for VICAS teleworking routers (not supported by the feature module)
PPP Modem Profile 1 ... 8	selects Modem Profile 1 ... 8 as configured in the [MODEM] menu
CAPI 1.1 EAZ 0 ... 9 Mapping	EAZ mapping for CAPI 1.1 applications

**Number** = the telephone number to use for this item.

**Mode** = the direction for matching the incoming telephone number (Called Party Number), either starting from the right (*right to left*, this is the default), or from the left (*left to right (DDI)*), only useful for the Direct Dial In (DDI) feature of point-to-point ISDN accesses (*Anlagenanschluß* in Germany).

### WAN Partner / Outgoing Calls

The [WAN Partner] [Advanced Settings] menu, where you configure ISDN partners, now contains a *Layer 1 Protocol* entry.

This entry also shows up in the **biboPPPTable**. The *Layer 1 Protocol* entry only has an effect on outgoing calls to this partner and on incoming calls which are identified by their calling party number. For an outgoing modem connection you should select one of the eight modem profiles.

The Layer 1 Protocol for incoming calls *not* identified by their calling party number—which will probably be the case for

most incoming modem connections, as they usually originate from the analogue telephone network, where no calling party numbers are supplied with the calls—is taken from the [*Incoming Call Answering*] settings.

The following table shows the possible values for the *Layer 1 Protocol* entry.



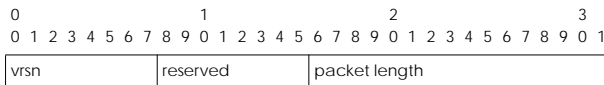
Note that most entries correspond to similar entries in the *Item* field of the [*Incoming Call Answering*] menu.

Value	Meaning
ISDN 64kbps	64kbps ISDN data connection
ISDN 56kbps	56kbps ISDN data connection
Modem	selects Modem Profile 1 as configured in the [ <i>MODEM</i> ] menu
DOVB	<u>d</u> ata <u>t</u> ransmission <u>o</u> ver <u>v</u> oice <u>b</u> earer; useful e.g. in the US where voice calls sometimes cost less than data connections
V.110 (1200 - 38400)	bit-rate adaption according to V.110 (1200 bps, 2400 bps, ..., 38400 bps)
Modem Profile 1 ... 8	selects Modem Profile 1 ... 8 as configured in the [ <i>MODEM</i> ] menu

## X.25 / TCP-TP0 Bridge

### Enhancements for RFC 1006 support

- The maximum packet length is now 65535 bytes including the header instead of just 8191 bytes.
- The transport packet header can now contain additional information for special X.25 packet types.



When transmitting packets where the *q bit is set (=1)*, *vrnsn* has to be set to **255** (instead of the default value 3).

For X.25 *reset request/indication* packets the *vrnsn* field is set to **255** and the *reserved* field is set to **1**.

For X.25 *reset confirmation* packets the *vrnsn* field is set to **255** and the *reserved* field is set to **2**.

A received X.25 *reset indication* packet contains 8 octets of data. The 4th octet contains the X.25 cause code and the 8th octet contains the X.25 diagnostic code. Data contained in transmitted reset packets is ignored.

*Reset confirmation* packets contain no additional data, so the packet length of such packets is always set to 4.

### Enhancements for RFC 1086 Support

X.25 on your BRICK now also supports extended addresses of address type=4 (standard addresses have type=3). Type 4 allows for variable length address fields and facilities.

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
address type										X.25 addr. length					X.25 address						
...										...					...						
...										...					...						
...										...					Protocol ID length						
Protocol ID										...					...						
CDUF length					Call User Data field					...					...						
...					...					...					...						
...					...					...					...						
...					...					...					...						
Facility length					X.25 Facilities					...					...						
...					...					...					...						

The address fields have the following meaning:

#### *address type* (2 octets)

A binary-encoded value in network order indicating the address type. The value 4 is used for extended X.25 addressing of this format.

*X.25 addr. length* (1 octet)

A binary-encoded value in network order indicating how many octets of the X.25 address there are.

*X.25 address*

The ASCII-encoded value of the X.25 address. Maximum length of 55 bytes allowed.

*Protocol ID length* (1 octet)

A binary-encoded value indicating the number of protocol ID octets there are.

*Protocol ID*

Meaningful at the remote system.

*CUDF length* (1 octet)

A binary-encoded value indicating the number of User Data octets there are.

*Call User Data field*

Meaningful at the remote system.

*X.25 Facility Length* (1 octet)

A binary-encoded value indicating the number of X.25 Facility octets there are.

*X.25 Facilities*

Meaningful at the remote system.

**Closed User Group support**

The Closed User Group (CUG) feature of X.25 is now supported. For this purpose the three variables *Cug*, *CugOutgoing*, and *CugBilateral* were added to both the **x25RouteTable** and the **x25RewriteTable**. The fields have the following meaning:

*Cug*      Closed User Group. Connections are only possible to DTEs in the same CUG. *Cug* can take values of -1 (*default*) if you do *not* want to use this feature, and 0-9999 for closed user groups.  
In the **x25RewriteTable** the value -2 is also possible,



meaning that the *Cug* field of a call request packet is cleared on rewriting.

### *CugOutgoing*

Closed User Group with outgoing access. Connections are also possible to DTEs not in the same CUG. *CugOutgoing* can take values of -1 (*default*) if you do *not* want to use this feature, and 0-9999 for closed user groups.

In the ***x25RewriteTable*** the value -2 is also possible, meaning that the *CugOutgoing* field of a call request packet is cleared on rewriting.

### *CugBilateral*

Bilateral Closed User Group. Connections are only possible between exactly two DTEs. *CugBilateral* can take values of -1 (*default*) if you do *not* want to use this feature, and 0-9999 for closed user groups.

In the ***x25RewriteTable*** the value -2 is also possible, meaning that the *CugBilateral* field of a call request packet is cleared on rewriting.

## New minipad options

To support the Closed User Group feature, three call options were added to the *minipad* application available on your BRICK.

*-c <cug>* Closed user group.

Possible values for *<cug>*: 0-9999.

*-o <outgocug>*

Closed user group with outgoing access.

Possible values for *<outgocug>*: 0-9999.

*-b <bcug>*

Bilateral Closed user group.

Possible values for *<bcug>*: 0-9999.

## Miscellaneous

- The *RewritingRule* in the ***x25RewriteTable*** can now take values of up to 999999 instead of 9999.

- The **debug x.25** command now outputs extensive information concerning layer 2 and layer 3 events on the existing links.

## Bugfixes

### CAPI

- When receiving a fax message the memory area allocated for that purpose was not returned to the system properly, resulting in less and less available user memory. This bug has been fixed.
- When you used the autoconfiguration feature of the ISDN interface (*Autoconfig=on* in the *isdnIfTable*) your CAPI configuration in the *capiConfigTable* was overwritten on each system start. This is no longer the case.

### Ethernet

Several problems regarding the ethernet driver were fixed.

- Occasionally duplicate packets were produced.
- The sender rarely ran into a blocked state.
- Under certain circumstances the ethernet response got very slow (e.g. ping reply times of up to 1000ms).

These problems no longer occur.

### HTTP

- The graphics on the BRICK's HTTP status page were sometimes not displayed correctly. They are now displayed correctly.

### IP

- You could not establish TCP connections to Windows PCs running Version 2.0 of the OnNet32 TCP/IP stack software of Ftp Software, Inc. This problem no longer occurs.

### PPP

- Leased lines: When changing or deleting bundle configurations in conjunction with leased lines the system occa-

sionally rebooted. After creating a new bundle it was necessary to reset the system to actually use it. These problems have been fixed.

- Multilink PPP: The *ifOutOctets* counter in the *ifTable* now displays the correct number of transmitted bytes.
- SPVs («Semi-permanente Verbindungen« of the german 1TR6 ISDN protocol) are now fully supported.
- VJHC: The Van Jacobson Header Compression now also works correctly in connection with Windows PCs.

#### V.42bis on Leased Lines

- When you used V.42bis data compression on a leased line, and the leased line failed, it could happen that after the leased line was back up the data connection could not be reestablished.  
This is no longer the case.

#### X.25

- Encapsulation x25/x25\_ppp: When using the minipad application in connection with this encapsulation the system sometimes rebooted after closing the minipad session.  
This problem was fixed.
- RFC 1086 Support: The Protocol ID Length field of X.25 packet headers is now correctly evaluated. This means that you can set the Protocol ID Length field to 0 and use the following 16 bytes for Call User Data.

