http://www.bintec.de

RELEASE NOTE
BIANCA/BRICK-XMP

December 18, 1998

New System Software:
Release 4.9 Revision 3

This document describes the new features, enhancements, bug-fixes, and changes to the BIANCA/BRICK-XMP System Software for Release 4.9 Revision 3.

NEW

Upgrading System Software

1. Retrieve the current system software image from BinTec's WWW server at http://www.bintec.de (Section: FTP Server).
2. With this image you can upgrade the BIANCA/BRICK-XMP 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 *BIANCA/BRICK-XMP User's Guide* under *Firmware Upgrades*.
3. Please note that since Software Release 4.9.1 there is a new update procedure in case there is not enough memory available to perform a software update via the **update** command from the SNMP shell.
   The new incremental update loads the new software image in blocks of 64 KB via TFTP and writes it to the Flash ROM immediately. Because this procedure offers no possibility to check the integrity of the image, please first use the option **"-v"** that verifies the image file. For more detailed information see New Update Procedure on page 12.
4. Once you've installed Release 4.9 Revision 3 you may want to retrieve the latest documentation (in Adobe's PDF format), which is also available from BinTec's file server at the address noted above.

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

What's New in Release 4.9.3

Release 4.9 Revision 3: Released: 18.12.98

Features: | Changes: | Bugfixes:

Features

TAF Syslog Messages

The TAF syslog messages have been extended.

Most of the syslog messages now additionally contain the IP address and the interface number.

For syslog messages concerning user authentification the syslog level INFO is used now.

New Timer in x25LinkPresetTable

To control the state of an X.25 connection and an X.25 partner in certain time intervals a new timer has been added to the ***x25LinkPresetTable***. The value of this timer can be set in the variable ***L2SupervTimer*** and is an integer between 100 and 30000, which is the value for the timer in milliseconds.

Changes

CAPI Syslog Messages

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

Examples for new syslog messages:

incoming call

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

outgoing call

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

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

CAPI DATA_B3_IND message

CAPI DATA_B3_IND messages now contain a valid datablk counter.

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

Bugfixes

Reboot when Establishing ISDN Connections

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

  ```
  PANIC: MIB getnext
  ...
  ```

  or

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

  This bug has been fixed.

X.25 Routing Priorities

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

This bug has been fixed by introducing a new timer in the variable ***L2SupervTimer*** in the ***x25LinkPresetTable*** described under New Timer in x25LinkPresetTable.

NAT on a Dial-Up Interface

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

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

  This bug has been fixed.

Dynamic IP Address Pools

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

RADIUS OSPF Interfaces

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

Accepting Calls with CAPI 1.1 Applications

- When an incoming call was accepted by a CAPI 1.1 application the Called Party Number was replied automatically as Connected Number to the caller. Some simple PABX could not handle this information and disconnected the call.

  Now the Connected Number isn’t sent by CAPI 1.1 applications any longer.

biboAdmCapiTcpPort/biboAdmTapiTcpPort

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

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

IPX: ripCircTable and sapCircTable

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

  This bug has been fixed.

IPX: Configuring the NetNumber

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

  This bug has been fixed.

What was New in Release 4.9.1

Release 4.9 Revision 1: Released: 30.10.98

Features: | Bugfixes: | Detailed Description:

Features

New BRICKware for Windows

With Release 4.9 Revision 1 also a new BRICKware for Windows is available, which must be installed to make use of the new CAPI User Concept.

New CAPI and TAPI Ports

BinTec product specific TAPI (TAPI concerns BinGO! Plus / Professional only) and CAPI ports have been officially registered by the IANA (Internet Assigned Numbers Authority) and have been changed as follows:

	OLD PORTS	NEW PORTS
CAPI	6000	2662
TAPI	6001	2663

This default values are only used, when BRICK and BRICKware are initially configured. It was necessary to introduce these changes, because in rare cases there occurred conflicts with applications, which used old CAPI and TAPI ports.
As a requirement for the operation of Remote CAPI/TAPI and the CAPI Tracer (PC) the values for the CAPI/TAPI ports configured on the BRICK and the PC must be the same.
A software update on the BRICK and on the PC does not change the configuration and with that also does not change the currently used port numbers. Therefore it is not necessary to change the ports after a mere update.
Nevertheless we recommend using the new ports. In the long term the new configuration will be necessary to resolve conflicts

that may occur with NAT and Firewall configuration. Please notice that a wrong configuration may be a potential source of errors.

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

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

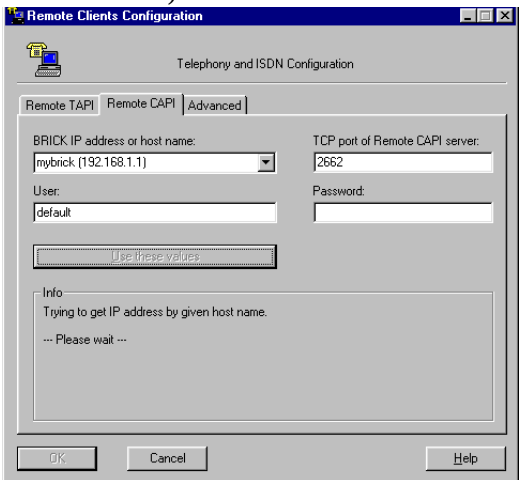
```
BRICK Setup Tool                          BinTec Communications GmbH
[IP][STATIC]: IP Static Settings                              mybrick

        Domain Name                       bricks.com
        Primary Domain Name Server        192.168.1.3
        Secondary Domain Name Server
        Primary WINS
        Secondary WINS
        Time Protocol                     TIME/UDP
        Time Offset (sec)                 0
        Time Update Intervall (sec)       86400
        Time Server                       192.168.1.3
        Remote CAPI Server TCP Port       2662
        Remote Trace Server TCP Port      7000
        RIP UDP Port                      520
        BOOTP Relay Server
        Unique Source IP Address

                  SAVE                    CANCEL
```

On the PC the CAPI/TAPI server ports are configured in the program “Remote Clients Configuration”. The CAPI Tracer of

the DIME Tools can be configured when starting a Trace session (Start / New CAPI Trace).



The current Unix Tools “capitrace”, “eft”, and “eftd” still use CAPI port 6000 as default setting. The ports of these programs can be changed by setting the environment variable “CAPI_PORT” under Unix. (e.g : CAPI_PORT=2662↵, export CAPI_PORT↵)

CAPI User Concept

A new CAPI User Concept has been implemented on the BRICK that gives you greater control of access to the BRICK’s CAPI subsystem. Each network user that attempts to access the BRICK’s CAPI subsystem must first be authenticated by using a user name and password which is configured on the BRICK (via the new ***capiUserTable***). Only if authentication is successful, the user can receive incoming calls or establish outgoing connections via the CAPI.

To take full advantage of the new CAPI User Concept mentioned above, a new BRICKware release has also been made available in Release 4.9 Revision 1. The most recent *BRICKware* version can always be retrieved from BinTec’s WWW server at http://www.bintec.de (Section: FTP-Server).

To allow for backward compatibility with older BRICKware releases, with Remote CAPI clients from other platforms, or with applications, which have implemented the Remote CAPI library a default user (Name = "default") has been defined. If the CAPI service is enabled for the default user, older BRICKware releases will still work with the new release. If disabled the CAPI service will be unavailable from older BRICKware releases. This also concerns the current versions of the Euro File Transfer Programs for Unix.

For information about configuring CAPI Users/Passwords see the Detailed Feature Descriptions on page 32

Setup Tool Menu Reorganization

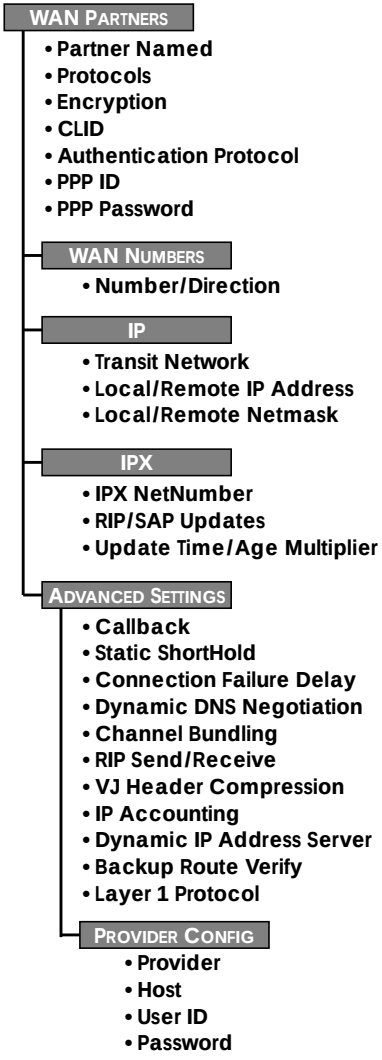
Beginning in Release 4.9 Revision 1, Setup Tool's Partner Management menus (contained in the WAN Partners section) have been reorganized to reflect a more logical structure. With the exception of several new configuration options (described below) these changes only affect the menu structure. Protocol 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 BRICK-XMP User's Guide which is available via BinTec's WWW Server.

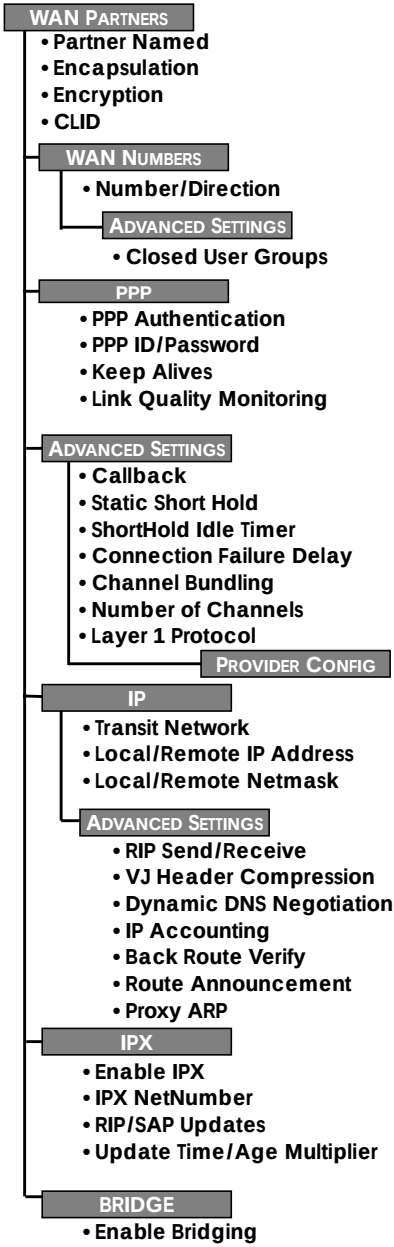
Also note that the new menu structure outlined below is also reflected in Setup Tool's X.25 → MPR →, FR → MPR →, and VPN → submenus.

Detailed information about new configuration options appearing in this release can be found in the section Detailed Feature Descriptions on page 32.

Old Setup Tool WAN Partner Menus



New Setup Tool WAN Partner Menus



Partner-Specific/non-Partner-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 34.

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 33.

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 4.9.1 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 BOOTmonitor.

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

1. Verifying the image

```
brick:> update -v tftpserver brk491.xp
Starting File Transfer..................... OK (754)
Checking new image... OK
File verifies OK
```

2. Updating the BRICK with the image

```
brick:> update tftpserver brk491.xp
Starting File Transfer .
Your current software release is 4.8.6.
New image has release 4.9.1.

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 successful verification of the file.

Transferring Configuration Files via the Serial Port

With Software Release 4.9.1 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 BIANCA/BRICK-XMP 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 the flash ROM of the BRICK.

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 BRICK's 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.

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

If you use cmd=put or cmd=state to transfer BRICK 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 BRICK (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.

Credits Based Accounting System

With dial-up WAN connections it may occur that charges rise, because of configuration errors. The Credits Based Accounting System gives BRICK administrators the ability to control charges. It allows the BRICK administrator to limit the number of connections, the connection time and the accounted charges of every subsystem during a specified period of time.

A detailed description of this feature you can find in “Detailed Features Description” under Credits Based Accounting System.

IP Route Announcement

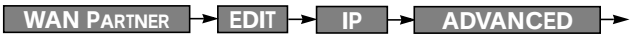
In the ***ipExtIfTable*** there is the new variable ***ipExtIfRouteAnnounce***, which adjusts for each interface under which conditions, in dependence of the ***ifOperStatus*** (***ifTable***) of the respective interface, the routes defined on this interface are propagated.

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

The variable can receive three possible values:

- ***up_only***
  The routes are only propagated, when the operational status of the interface is up.
- ***up_dormant***
  The routes are only propagated, when the operational status of the interface is up or dormant.
- ***always***
  Independent from the operational status of the interface the routes are always propagated. If e.g. a dial-up interface is in the state “blocked”, the route is propagated.

In Setup Tool the configuration is made under




```
BIANCA/BRICK-XMP Setup Tool                    BinTec Communications GmbH
[WAN][ADD][IP][ADVANCED]: Advanced Settings ()                       brick


RIP Send                                       none
RIP Receive                                    none

Van Jacobson Header Compression                off
Dynamic Name Server Negotiation                yes

IP Accounting                                  off
Back Route Verify                              off
Route Announce                                 up or dormant
Proxy Arp                                      off

                    OK                                CANCEL

Use <Space> to select
```

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

up_only specifies that the routes are only propagated, when the operational status of the interface is up.

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

Proxy ARP

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

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

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

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

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

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

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

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

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

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

The requirements for an answer to a ARP request from the LAN by the BRICK are that the destination address would be routed to a different but the LAN interface and that on both interfaces (LAN and destination WAN interface) proxy ARP is turned on (***on*** for the LAN interface and ***on*** or ***up_only*** for the respective WAN interface). Beyond that the ***ifOperStatus*** of the WAN interface must have the demanded state.

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

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

Access Lists

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

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

X.25 PAD

The PAD is a data assembly / disassembly facility used to connect character-oriented asynchronous data terminal equipment (DTE) to the packet-oriented X.25 network (Datex-P). It is the task of PAD to convert character streams coming from the DTE into data packets and resolve data packets coming from the network into individual character streams that can be displayed on the DTE. In this context the character-oriented data terminal equipment is also called start-stop mode DTE (short: DTE) and a remote X.25 host is defined as packet mode DTE.

Recommendation X.29 defines the procedures between a PAD and a packet-mode DTE or another PAD and recommendation X.28 defines the DTE interface of a start-stop mode DTE accessing the PAD.

The PAD functionality has been implemented in the software of the BRICK with this Software Release and can be used by all customers, who have licensed X.25.

A detailed description of the X.25 PAD and its configuration can be found in the current version of the Extended Features Reference, which can be retrieved from BinTec's FTP server at http://www.bintec.de.

X.25 Dialout Without Configuration

In an X.25 network there are often a lot of different connection partners that cannot all be configured on the BRICK or even on different BRICKs. In addition there are often so many X. 25 part-

ners that a configuration is not possible because of the limited size of the Flash ROM of the BRICK.

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

For this feature two new values for X.25 encapsulations have been added. The variable ***Encapsulation*** in the ***biboPPPTable*** and the corresponding item **Encapsulation** in Setup Tool's WAN PARTNER/ADD menu now also can receive the value ***x25_noconfig*** (Setup Tool: **X.25 No configuration**) and ***x25_noconfig_nosig*** (Setup Tool: **X.25 No configuration, No Signalling**).

The value ***x25_noconfig*** uses an X.25 specific signalling in the D-channel for the data call.

The value ***x25_noconfig_nosig*** is an variation of the value ***x25_noconfig*** and uses in contrast to "X.25 No Configuration" an ISDN specific signalling in the D-channel for the data call.

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

Pools for Dynamic IP Address Assignment

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

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

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

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

WINS (NBNS) Negotiation over PPP

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

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

DHCP Server Functionality

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

A detailed description of this new feature can be found in BinTec's Software Reference in Chapter 7 under the heading "DNS and WINS Relay". The Software Reference can be retrieved from BinTec's FTP server at http://www.bintec.de (Link: FTP Server). There you can find the respective link under "Reference Manuals" on the respective product page.

X.25 in Setup Tool

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

X.25 → LINK CONFIGURATION →

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

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

X.25 → ROUTING → ADD →

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

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

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

Changes

TCP Optimization

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

Configuration: State File

When writing a state file with cmd=state the following variables are not output respectively substituted by "****" with Software Release 4.9.1:

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

CAPI: PLCI and NCCI

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

ipExtIfKeepalive

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

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

biboAdmBrdType

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

Example for BIANCA/BRICK-XMP:

```
inx  Slot(*ro)              Type(ro)          HWRelease(ro)   FWRelease(ro)
     PartNo(ro              Connector(rw)

00   0                      "BRICK (2/8 MB)   "3.1"           "2.6"
     "BIANCA/BRICK-XMP"     auto
```

In this example (1/4 MB) stands for: 1 MB flash and 4 MB CPU-DRAM.

Charging Information

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

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

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

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

TotalUnits in the ***biboPPPStatTable*** and the variable ***biboPPPLinkUnits*** in ***biboPPPLinkTable***.

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

Example:

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

Bugfixes

isdnLoginOnPPPDispatch

- When the variable ***isdnLoginOnPPPDispatch*** (***isdnTable***) is set to allow, incoming ISDN calls with the service indicator "telephony" should be routed to the ISDN login daemon, even though the call via the ***isdnDspLocalNumber*** has the matching service "PPP" in the ***isdnDispatchTable*** (for non-PABX products).

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

  This bug has been fixed.

LAPB Encapsulation with Compression

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

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

biboPPPLQMTable

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

  This bug has been fixed.

B-Channel Bundling with RADIUS

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

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

This bug has been fixed.

FM-8MOD

- The BRICK-XMP sometimes refused (automatic baud-negotiation was unsuccessfull) incoming modem connections from V.90 compatible modems. Often times a second or third connection attempt, from the same calling equipment, was subsequently successful. 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.

CAPI

- Direct-Dial-In with CAPI Applications

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

  For this purpose some applications only interpret those digits, which are signalled to them with the “INFO_IND” message and ignore the digits in the “CONNECT_IND” message. These digits of the “CONNECT_IND” message were not send additionally in an “INFO_IND” to the application.

  Especially, when the Called Party Number was received completely in one message by the BRICK, the application did not get any “INFO_IND” message and it occured that in such a case a call was incorrectly accepted or not accepted.

  This bug has been fixed.

  Now all digits of the Called Party Number, which are contained in the “CONNECT_IND” message are additionally sent in an “INFO_IND” message.

- Data Transfer in Transparent Mode

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

  This bug has been fixed.

TAF

- Reestablished Dial-Up Connections

  Dial-up connections can be closed automatically with the shorthold mechanism of the BRICK. When the partner is still authenticated with TAF and the connection is established again, the BRICK checks automatically with a short authentication, if the partner is still the same, before data can be transferred. In this case you could transfer data for

5 seconds until the BRICK noticed the short authentification was not successful.

This bug has been fixed.

- ipTafTable

  Entries in the ***ipTafTable*** with the ***State*** value ***authentificating*** sometimes were not deleted automatically.

  This bug has been fixed.

- RADIUS with TAF

  Activating TAF together with RADIUS interfaces is now supported.

- Backup ACE Server

  Up to now an automatical shift to a backup ACE server was not possible. Now operation with a backup ACE server is supported.

- Authentification Repetition

  180 seconds (= 3 minutes) before TAF authentification (lifetime) expires, the user is requested to authenticate again. Up to now this was only initiated, when the value of the variable ipExtIfAuthLifeTime (ipExtIfTable) could be divided by 10 or when the user manually had set the variable iptafTimeout (ipTafTable) to a value that could be divided by 10.

  This bug has been fixed in the current release.

- Authentification by ACE Server

  In rare cases it occured in connections with a complex configuration of IP routes that the ACE server did not accept the authentification of a user, although the user password was correct.

  This bug was dependent on the order of the entries in the ipRouteTable and has been fixed with this Software Release.

Network Address Translation

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

  This bug has been fixed.

ISDN: Alcatel 4200

- When a BRICK connected to an Alcatel 4200 ISDN PABX tried to set up an outgoing call, a reboot occured. This happened because charging information is transmitted in currency amounts by the Alcatel 4200 and not in units.

  This bug has been fixed.

localUdpAllowTable

- When the ***localUdpAllowTable*** contained more entries than the ***localTcpAllowTable***, there sometimes occured a reboot of the BRICK.

  This bug has been fixed.

Call Collisions with MS Callback

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

  This bug has been fixed.

ifconfig Command

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

  This bug has been fixed.

  Now you can use e.g.

  ifconfig en1 168.1.1.1 netmask 255.255.255.0 up

  to configure the IP address.

Setup Tool: WAN Partner

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

  This bug has been fixed.

Setup Tool: Access Lists

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

  This bug has been fixed.

Known Bug

- B-Channel Trace

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

Detailed Feature Descriptions

CAPI Users/Passwords

The new CAPI→ menu has been added to Setup Tool for configuring CAPI users and passwords. Configuring CAPI Users is seemingly straight forward; select ADD in the CAPI→ USER→ submenu (shown below) to add/modify CAPI users

```
BRICK-XMP Setup Tool                          BinTec Communications GmbH
[CAPI][User]: Configure CAPI USers                                brick


        Name            Password        CAPI
        default                         enabled


        ADD             DELETE          EXIT
```

If the ***capiUserTable*** is empty at boot time, a default entry (as shown above) is automatically added. The default user is enabled and no password is required.

In the subsequent ADD menu define the following fields:

Name specifies the user name (up to 16 characters) to enable/disable CAPI access for.

Password specifies the password this user must authenticate with when accesing the CAPI subsystem.

CAPI determines whether the CAPI service is “enabled” or “disabled” for this user.

New Setup Tool PPP Configuration Options

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.

```
BRICK-XMP Setup Tool                          BinTec Communications GmbH
[PPP]: PPP Profile Configuration                                  brick

        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. Corresponds to the variable ***biboPPPProfileAuthProtocol*** in the ***biboPPPProfileTable***.

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. This item corresponds

to the variable ***biboPPPProfileAuthRadius*** in the ***biboPPP-ProfileTable***.

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***. Corresponds to the variable ***biboPPPProfileLQMonitoring*** in the ***biboPPPProfileTable***.

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 and successfully negotiated with the peer, the BRICK sends LCP echo requests to the remote partner every three seconds. After five unanswered requests the PPP interface's ***ifOperStatus*** is set to "down" for leased lines ("dormant" for dial-up connections). 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 BRICK to gather PPP Link Quality statistics for a specific PPP partner. When enabled, link statistics are continously written to the BRICK's ***biboPPPLQMTable,*** when a connection is established with this partner.

Credits Based Accounting System

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

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

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

Setup Tool Menus

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

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

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

ISDN → **Credits** →

```
BIANCA/BRICK-XMP Setup Tool                 BinTec Communications GmbH
[ISDN][CREDITS]: Configure Credits                              mybrick


Select Subsystem

Subsystem           Surveillance
capi                off
ppp                 off
isdnlogin           off

EXIT


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

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

capi

ppp

isdnlogin

ISDN → Credits → ppp

```
BIANCA/BRICK-XMP Setup Tool                    BinTec Communications GmbH
[ISDN][CREDITS][EDIT]: Configure ppp Credits                      mybrick

Surveillance                                                      on
Measure Time (sec)                                                86400
Maximum Number of Incoming Connections                            on
                                                                  2
Maximum Number of Outgoing Connections                            on
                                                                  20
Maximum Charge                                                    off
Maximum Time for Incoming Connections (sec)                       on
                                                                  28800
Maximum Time for Outgoing Connections (sec)                       on
                                                                  28800

                    SAVE                                                CANCEL

Use <Space> to select
```

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

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

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

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

Maximum Number of Outgoing Connections = The number of allowed outgoing connections during the measure time. If you set it on you can enter an integer from 0 to 2147483647. Default value is 100 calls.

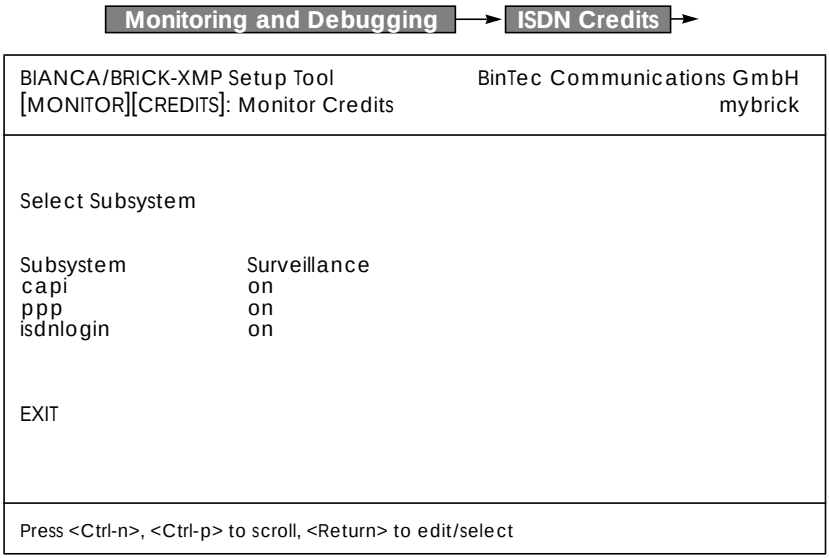
Maximum Charge = The maximum allowed charge information during the measure time. If you set it on you can enter an integer from 0 to 2147483647. Default value is off.

Charge information is measured in units or when charge information is sent as currency amounts, the charge is measured

in 1 / 1000 of the respective currency. (E.g. receiving charging information “0.12 DM” would result in a value of 120 charging units.)

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

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



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

Monitoring and Debugging → ISDN Credits → ppp

```
BIANCA/BRICK-XMP Setup Tool                    BinTec Communications GmbH
[MONITOR][CREDITS][STAT]: Monitor ppp Credits                    mybrick

                                    Total     Maximum   % reached

Time till end of measure interval (sec)  7794    86400     91

Number of Incoming Connections           0       2         0
Number of Outgoing Connections           0       20        0

Time of Incoming Connections             0       28800     0
Time of Outgoing Connections             0       28800     0

Charge                                   0

EXIT
```

Here you can see the current values.

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

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

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

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

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

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

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

IP Address Pools

Pool ID Selection

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

1. **Assigning a Static IP Address**
   When there exists an entry in the ***ipRouteTable*** for the dial-in client, where ***ipRouteMask*** is set to a host route (= ***255.255.255.255***) and ***ipRouteType*** has the value ***direct***, in this case the IP address stored in the variable ***ipRouteDest*** of this routing entry is taken to be assigned for this WAN partner.
   If caller can't be authenticated locally via the MIB, RADIUS server(s) are contacted. If a server authenticates the caller, and there is a User-Record entry
   BinTec-ipRouteTable="ipRouteMask=***255.255.255.255***
   ipRouteType=direct
   ipRouteDest= x"
   the IP address stored in the variable ***ipRouteDest*** of this entry is taken to be assigned for this WAN partner.
2. **Assigning an IP Address from an Address Pool**
   When the procedures described under 1. were not successful, the IP address is assigned from the Pools.
   Once the caller is identified (either inband or outband), the WAN partner's ***biboPPPTable*** entry is compared. If the ***IPAddress*** field = "dynamic_server" AND an address is available from the pool identified by the ***PoolId*** field, then a free address is assigned.
   If caller can't be authenticated locally via the MIB, RADIUS server(s) are contacted. If a server authenticates the caller and there is a User-Record entry BinTec-biboPPPTable="biboPPPIpAddress=dynamic_server", the pool ID is determined from the User-Record entry BinTec-biboPPPTable="biboPPPIpPoolId=x".

MIB Tables Overview

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

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

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

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

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

Example Configuration of IP Address Pools via Setup Tool

A. Dial-In Partner without RADIUS

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

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

Pool ID	1
Number	10.5.5.5
Number of Consecutive Addresses	5

WAN PARTNER → ADD **Create Partner Interface**

Here you'll need to set:

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

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

IP Transit Network	dynamic_server

In the ADVANCED SETTINGS submenu define the Pool ID

IP Address Pool	1

B. Dial-In Partner with RADIUS server

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

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

Pool ID	2
Number	192.168.80.20
Number of Consecutive Addresses	20

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

BinTec-biboPPPTable=”biboPPPIpPoolId=2”

Example Configuration of IP Address Pools via SNMP Shell

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

A. Dial-In Partner without RADIUS

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

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

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

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

B. Dial-In Partner with RADIUS server

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

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

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

   BinTec-biboPPPTable=”biboPPPIpPoolId=2”

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

Important Note:

Overlapping Address Pools:

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

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

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

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