http://www.bintec.de

RELEASE NOTE
BinGO!

December 5,1997

System Software:
Release 4.6 Revision 4

This document describes the new features, enhancements, and changes to the BinGO! System Software that could not yet be included in the printed manual.

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 BinGO! with the **update** command from the SNMP shell via a remote host (i.e. using telnet, or isdnlogin) or by using the **BOOTmonitor** if you are logged in directly on the console.
   Information on using the BOOTmonitor can be found in the *BinGO! User's Guide* under *Firmware Upgrades*.
3. Once you've installed Release 4.6 Revision 4 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 FTP server.

What's New in Revision 1

Release 4.6 Revision 4: Released: 05.12.97

Features

Support of up to 8 MAC Addresses

With Release 4.6 Rev. 4 BinGO! supports up to 8 LAN addresses instead of four. This feature is free of charge and will work by simply upgrading to 4.6.4.

Microsoft Compatible CHAP (MS-CHAP)

Microsoft's proprietary CHAP implementation (Microsoft PPP CHAP Extensions, Revision 1.3) is now supported. PPP partners that use MS-CHAP authentication should now be configured by setting ***biboPPPAuthentication*** to **`ms-chap`**, or by selecting MS-CHAP in the "PPP Authentication Protocol" field in Setup Tools's [WAN Partners] menu.

Another encapsulation method (**`all`**) has also been added to ***biboPPPAuthentication*** to allow MS-CHAP (as well as CHAP and PAP) authentication to be attempted. As of Revision 4 the following authentication options my be configured.

biboPPPAuthentication	Attempted Authentication Methods[a]
`none`	No inband authentication is performed.
`pap`	Authentication ONLY via PAP.
`chap`	Authentication ONLY via CHAP.
`both`	Authentication via CHAP OR PAP.
`radius`	Authentication via biboAdmRadiusServer using CHAP OR PAP.
`ms-chap`	Authentication ONLY via MS-CHAP.
`all`	Authentication via CHAP, MS-CHAP, OR PAP.

a. In the case of multiple protocols authentication is attempted in the order mentioned.

Note: Most older RADIUS implementations do not support MS-CHAP authentication.

Configurable PPP Defaults

With Revision 4 the default settings used for dial-in PPP connections are configurable using the new ***biboPPPProfileTable***. These settings only apply to incoming connections from callers that can't be verified via CLID (Calling Party's Number).

The new ***biboPPPProfileTable*** consists of three fields.

Name(*ro)	–Read-only; the profile name.
AuthProtocol(rw)	–Defines the default protocol to use for PPP authentication.
AuthRadius(rw)	–Controls the use of a RADIUS server when default PPP settings are applied.

See Default PPP Handling for detailed information on how the default PPP settings are determined for incoming PPP connections.

Auto-Logout Timer

A new Auto-Logout timer has been implemented on the BinGO!. This feature means that each login session started on the BinGO! now automatically times out after a pre-configured timer expires (by default 900 seconds).

This feature has been added for improved security and better cost management. Unattended login sessions are less available to unauthorized parties. By closing all connections started by the login session unwanted connection costs are minimized.

Note that if you do not manually issue the t command with a different time value your login session will be terminated 900 seconds (i.e. 15 minutes) after the last user input (i.e. keystroke). The timer does not care what kind of application is running at

the time or whether or not a data transfer is in progress, so in case you expect a long, non-interactive terminal session (setup tool monitoring, ISDN trace session, very large data transfer, etc.) you should disable the timer.

See The t Command on page 11 for information on using and changing the Auto-Logout timer.

SetupTool Search Functionality

Setup Tool lists are now automatically sorted alphabetically and can be easily navigated using a new search mechanism. For information about how list searching works see the section Searching Menu Lists in Setup Tool on page 11.

This feature is mainly intended for sites with many partner interfaces.

Delayed Callback

The PPP Callback feature on the BinGO! has been extended. An additional value (delayed) has been added for the answering side that delays the callback for the amount of time (in seconds) configured in the *RetryTime* variable of the ***biboPPPTable***.

The following table gives an overview of the Callback options currently available:

Setup Tool	SNMP Shell	Explanation
no	disabled	no Callback possible
expected (awaiting callback	expected	wait for a call back from a partner
yes	enabled	accept callback requests and call back immediately
yes (delayed)	delayed	accept callback requests and call back after *RetryTime* seconds
yes (PPP negotiation)	ppp_offered	accept callback requests and negotiate the callback number inband (see section New PPP Callback Method for details)

Note that *delayed* callback currently only works for calls identified outband by their CLID.

DNS Negotiation Configuration

DNS Negotiation can now be configured on a per partner basis allowing you to better control how (and from which partners) the BinGO! will negotiate DNS settings.

Beginning in Release 4.6 Revision 4 each Dial-Up partner can be separately configured so that the BinGO! either:

1. Accepts DNS settings from the partner.
2. Offers DNS settings to this partner.
3. Does not negotiate DNS settings with the partner.

See the section Partner-Specific DNS Negotiation on page 13 for information on configuring partner-specific DNS negotiation settings.

Extended DHCP Support

Some RFC 2131 based DHCP clients use DHCP messages that are not compatible with the RFC 1541 conformant DHCP Server on the BinGO!. This means that the BinGO! was unable to reject requests from DHCP clients on a different subnet than the BinGO!. Problems occured e.g. when moving a laptop PC from a LAN with a Windows NT-based DHCP server to a different LAN with a BinGO! as DHCP server—the laptop then did not accept the new IP address offered by the BinGO!.

The BinGO! now supports RFC 2131 based clients, solving the problem.

CAPI Extensions

CAPI 2.0 and 1.1

The CAPI_GET_VERSION and CAPI_GET_SERIAL_NUMBER messages can now be used under CAPI 1.1 and CAPI 2.0 to retrieve hardware and/or software information about the BinGO!.

CAPI_GET_VERSION

Returns the BinGO!'s current software version, (retrieved from the ***biboAdmSWVersion*** object).

CAPI_GET_SERIAL_NUMBER

Returns serial number information.
The information consists of a 7 character string that identifies the product type and SystemID as follows:

Digit 1: Identifies the product:

4 = BRICK-XS, BRICK-XS Office, V!CAS
5 = BRICK-XM
6 = BRICK-XL
7 = BinGO!

Digits 2-7: The last 6 digits of the BinGO!' serial number. (From the ***biboAdmSystemId*** object).

CAPI 2.0

The B2 protocol 12, "LAPD according to Q.921 including free SAPI selection." is now supported on the BinGO!.

X.31 in D-Channel

X.31 in the D-Channel on the BinGO! is now officially certified by the German BZT (*Bundesamt für Zulassungen in der Telekommunikation*) .

Bugfixes

PPP

- The authentication entry of a WAN partner was mistakenly set to **radius** when a second call came in and multilink PPP was not activated.
  This bug has been fixed.
- Partner specific local PPP identification (***biboPPPLocalIdent***) can now be used in connection with any authentication protocol (***biboPPPAuthentication*** = CHAP, MS-CHAP, or PAP).

In previous releases there was a limitation involving callback negotiation and inband CHAP authentication.

- As of release 4.6 Revision 4 the BinGO! supports Cisco's proprietary (non RFC 1974 conformant) packet format for compressed (STAC) PPP datagrams. In some versions of Cisco software CCP negotiation is repeated periodically even when the connection and CCP has already been established. Prior to revision 2 this behaviour resulted in a memory leak on the BinGO!.

CAPI 1.1 and 2.0

- Under certain conditions the BinGO! did not create a »Sending Complete« indication. This bug has been fixed.
- In previous releases, CAPI applications that were initially notified of an incoming call (CONNECT_IND) but did not actually receive the call were incorrectly sent a "Normal call clearing" (DISCONNECT_IND) message. This has been corrected. Applications now receive an "Another application got the call." message (CAPI 2.0) or an "Abort D-Channel layer 1" message (CAPI 1.1).
- The BinGO! sometimes rebooted when the CAPI and TAPI port numbers were exchanged or when large amounts of data were transmitted over a telnet session to the CAPI port. This has been corrected.
- When the ***isdnDispatchTable*** and ***isdnloginAllowTable*** are empty the BinGO!'s CAPI server cannot distribute incoming calls because the isdnlogin service accepts the call. In previous releases CAPI applications were falsely notified of the call and then received a DISCONNECT_IND (because the call was already given to the isdnlogin service). CAPI applications are no longer notified of the call in such cases.
- When a CAPI application was started the message: "INVALID MESSAGE LENGTH!" was improperly displayed in a running capitrace session. This problem has been corrected.

CAPI 2.0

- If during a fax transmission the remote side is no longer responding, the CAPI application is sent the error message "CAPI2_E_FAX_REMOTE_PROCEDURE_ERROR" instead of a "CAPI2_E_FAX_REMOTE_ABORT" message as in previous releases.
- In previous releases the CAPI 2.0 "FACILITY_REQ" message was not transmitted when the "Facility Request Parameter" field exceeded 6 bytes.

TFTP put Transfers

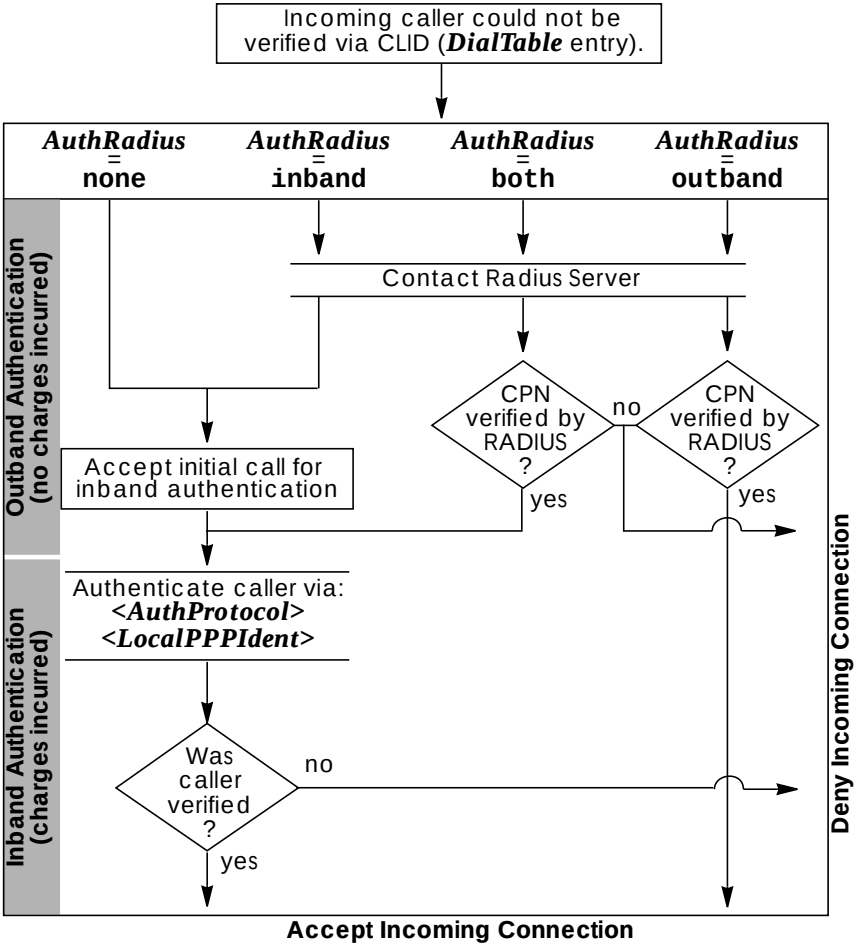
- When transferring files to remote hosts via the TFTP "put" command signed variables were sometimes transmitted as unsigned. As of revision 2 this no longer occurs.

Detailed Feature Descriptions

Default PPP Handling

The diagram below shows how the BinGO!'s default PPP settings are determined using the settings found in the ***biboPPPProfileTable***.

Note: These settings ONLY affect the default handling of INCOMING connections that couldn't be identified by the Calling Party's Number.



The t Command

The **t** command has been added to the SNMP shell and defines the number of seconds to wait (once terminal input is idle) before closing the current login session. When the BinGO! closes the login shell, all programs (setup session, trace, etc) started during the session that are currently running are also closed.

Usage:

Each time a user logs in the timeout is set to 900 seconds by default. To change the timeout setting ente:

t <*seconds*>

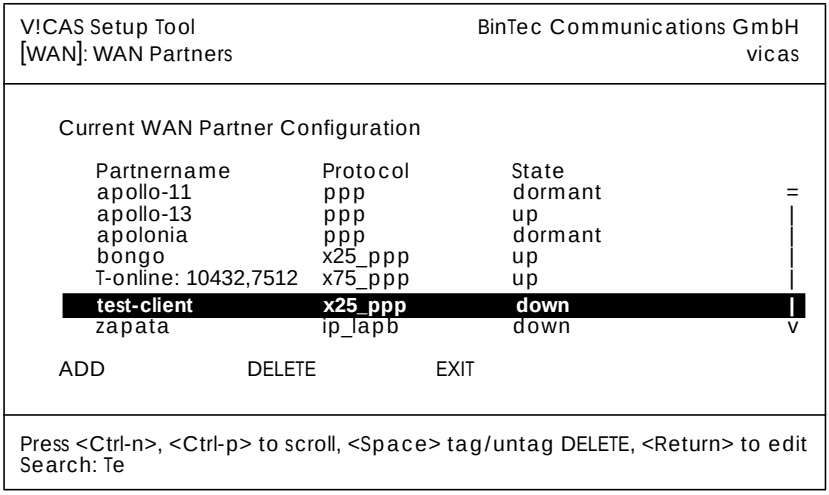
The auto-logout feature can be disabled completely (for the current login session only) by setting the timer to 0.

NOTE: This feature is primarily intended for security / cost-control reasons. If you expect a long, non-interactive terminal session (setup tool monitoring, ISDN trace session, etc.) you should disable the timer.

Searching Menu Lists in Setup Tool

Setup Tool menus that display list entries are sorted alphabetically using the contents of the first field.

To search menu list items enter a valid search character (only printable characters). The cursor automatically jumps to the first match in the list. As long as the search is active subsequent characters entered are appended to the search string. The current search string is shown in the bottom portion of the terminal window. Entering a non-printable character resets the current search (and possibly performs an action; e.g. tab, space, etc.). The <backspace> key (and possibly <delete> depending on terminal settings) can be used to edit the search string. Search characters are case-insensitive (Entering the letter “t” matches both “t” and “T” characters).



Assuming the above [WAN Partners] menu list the following key sequences would have the following effect:

Key Sequence	Resulting Effect
t, or **T**	Cursor jumps to the: **T-Online 10432,7512** entry.
te, **TE**, **tE**, **Te**	Cursor jumps to the: **test-client** entry.
a p o l o	Cursor jumps to: **apollo-11** entry first then to: **apolonia** after the last "o".

Note also that a search can only be performed when the cursor is in a list field (and not when in an ADD, DELETE, EXIT, CANCEL, or SAVE field).

Partner-Specific DNS Negotiation

Together the MIB variables ***biboPPPIpAddress*** and ***biboPPPDNSNegotiation*** control how DNS negotiation is handled with the respective PPP partner.

biboPPPDNSNegotiation

The type of negotiation to perform with this client. Default value: “enabled”. Possible values include:

`disabled`(1), `enabled`(2), `dynamic_client`(3), `dynamic_server(4)`

biboPPPIpAddress

The type of IP address for this dial-up partner. Possible values include:

`static`(1), `dynamic_server`(2), `dynamic_client`(3)

The table below illustrates the effect of using theses two variables to control DNS negotiation.

Variable:Setting:	Negotiation Handling:
Negotiation=disabled	No negotiation is performed.
Negotiation=enabled **AND** ***IpAddress***=dynamic_client	Remote side is always asked for primary/secondary (***biboAdmNameServer***/***biboAdmNameServ2***) server. Local values are always overwritten is offered by remote.
Negotiation=enabled **AND** ***IpAddress***=dynamic_server **OR** ***IpAddress***=static	Values for primary/secondary server are only sent if requested by remote side and they are configured.
Negotiation=dynamic_client	Remote side is always asked for primary/secondary nameserver addresses.
Negotiation=dynamic_server	Values for primary/secondary servers are sent only when requested by remote side.

4.6 Revision 1: Released: 01.10.97

Features

New charge-dependent Short Hold

You can now configure a charge-dependent Short Hold on your BinGO!.

For a detailed description of this feature and its configuration please refer to page 20.

New PPP Callback Method

With release 4.6.1 an additional Callback method according to RFC 1570, named *ppp_offered*, is available.

For a detailed description of this feature and its configuration please refer to page 23.

BRICKware for Windows

- The »Reflection« TCP / IP stack for Windows from WRQ is now also supported by the BinTec CAPI2032.DLL.
- The ISDN trace of Dime Tools now also decodes LCP extensions and IPCP extensions.
- Improved Time Server features; The Dime Tools Time Server is now also capable of supplying your BinGO! with its local time.

  You can now choose between Greenwich Mean Time (GMT) and Local Time in the Configuration–Time Server menu of Dime Tools.

  We recommend that you use the Local Time setting of Dime Tools and set Time Offset (seconds) to 0 in the BinGO!'s [*IP*][*Static Settings*] Setup Tool menu. This will also automatically take care of daylight savings time[1].

  Note, however, that this method is no longer fully compatible to RFC 738.

1. Called »Sommerzeit« in Germany.

If you select Greenwich Mean Time, the current setting for the TZ (timezone) environment variable in the AUTOEXEC.BAT will be displayed on the screen. For Germany this will usually be

set TZ=GST1GDT

Please refer to your PC documentation for a description of the TZ entry.

DHCP Server

When using dynamic IP address assignment you can now associate an IP address with a MAC (hardware) address. This can be useful for small networks which shall be managed from a central server.

In Setup Tool this can be done in the [*IP*] [*DHCP Server*] [*ADD*] menu. Set the *Number of consecutive addresses* to **1** and enter the appropriate *MAC Address*.

ifstat Command

The *ifstat* command now takes the following three optional parameters:

-l Displays the full length of the interface descriptions (normally the description is only displayed up to the 12th character).

-u Only display information on interfaces which are in the **up** state.

‹ifcname›
Only display information on interfaces whose description starts with the given characters (e.g. **ifstat en1** will display information on the interfaces en1, en1-llc, and en1-snap).

IP

New ipPriorityTable

We added the ***ipPriorityTable*** to the MIB to be able to make routing protocol priorities user-definable.

This can be useful when you want your BinGO! to prefer certain routing protocols over others.

Each table entry consists of two variable, Proto, which specifies the routing protocol, and Value, which specifies the priority for this protocol. Value can be 0 to 63, where lower numbers define a higher priority, i.e. 0 is the highest priority, 63 the lowest.

The available routes with the highest priorities are always used when routing IP packets.

The following route types are supported at the moment:

direct Routes where *ipRouteType*=**direct**, which define IP addresses for interfaces.
Default priority: 0

static Routes where *ipRouteType*=**indirect**, which were not generated by a routing protocol.
Default priority: 0

rip Routes which were learned by the RIP protocol.
Default priority: 20

New ipExtIfTcpCksum Switch

The ***ipExtIfTable*** now contains the variable *ipExtIfTcpCksum* which can be used to enable (**check**) or disable (**dont_check**) TCP checksumming.

We recommend setting this switch to **dont_check** only on LAN interfaces for performance critical CAPI applications.

Do *not* use **dont_check** for interfaces where Van Jacobson Header Compression is enabled, or over which other routers can be reached, because in these cases the correct function of the TCP cannot be guaranteed.

For traffic inside your LAN you can disable (**dont_check**) the TCP checksum, because ethernet and token ring have their own CRC checksums to ensure data integrity.

IP Performance Optimization

IP performance was optimized, especially for Remote CAPI applications.

IP Routing

While setting up a dialup interface (ISDN) for IP traffic other routes to the same IP destination are used until the dialup interface has reached the **up** state. This ensures a smooth transition from an alternative route back to the main route.

In previous releases IP traffic was blocked for the duration of the interface setup.

ISDN

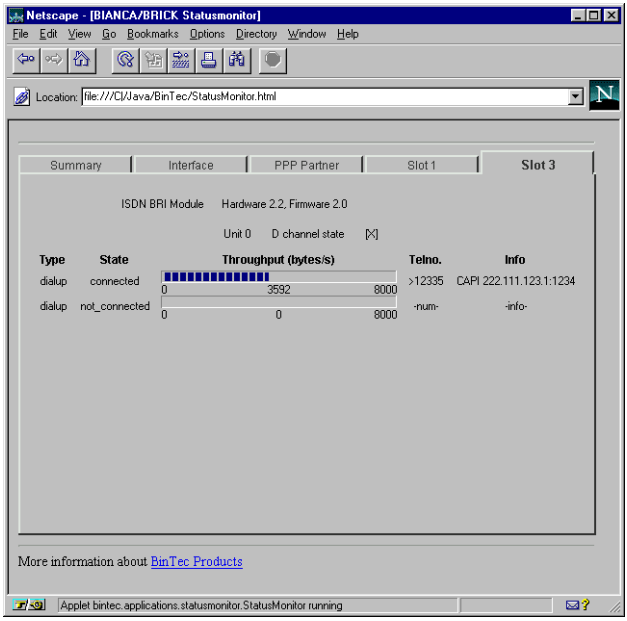
- A Syslog message is now created each time the isdnlogind takes an incoming call.
- A new variable, *DialOutPrefix*, was added to the ***isdnStkTable***. The contents of this variable are appended to the callback number when using PPP inband callback (Microsoft CBCP, see page 25).

New Java Status Monitor

The Java Status Monitor allows you to monitor your BinGO! router dynamically and see:

- how long the system has been running
- current traffic for each interface
- connection information (which partners are connected, duration of call, accumulated costs, …)
- number of B-channels currently in use, and details for each connection

You can choose from the BinTec ISDN Companion CD setup whether you want to install the Java Status Monitor.



For further information please refer to the STATMON.TXT and DEVELOP.TXT files installed with the Java Status Monitor.

You can also download the latest version of these files, along with the complete Java Status Monitor package from our file server in the *ftp://ftp.bintec.de/pub/brick/statusmon* directory.

PPP

Local PPP ID for each WAN Partner

You can now specify a different Local PPP ID for each WAN partner. This ID is stored in the *biboPPPLocalIdent* variable in the ***biboPPPTable***. If you do not specify a special Local PPP ID for a partner, the default PPP ID is taken from the *biboAdmLocal-PPPIdent* variable in the ***admin*** table.

In the Setup Tool you can configure the Local PPP ID field in the [*WAN Partner*][*ADD*] menu.

MTU Negotiation

The remote MRU / MRRU (Maximum Receive Unit, maximum packet length) negotiated by the LCP (Link Control Protocol) will now be written to the *ifMtu* variable of the corresponding ***ifTable*** entry to avoid packet discarding by peer caused by exceeding the negotiated peer's MRU / MRRU.

RIP

RIP performance was improved. DEBUG level syslog messages are now also generated when routes are created or deleted.

Setup Tool

DHCP Server

You can now specify a MAC address when configuring your BinGO! as a dynamic IP address server (see section DHCP Server above).

Local PPP ID

You can now specify a different Local PPP ID for PAP / CHAP authentication for each WAN Partner (menu [*WAN Partner*] [*ADD*]).

System

At each system start a syslog message will now be generated.

"system ‹sysname› started at ‹time›"

Detailed Feature Descriptions

Charge-dependent Short Hold

The Short Hold timer available in previous releases could be used to disconnect a dialup call after a configurable, but fixed, period of inactivity.

As ISDN calls are normally not charged according to the exact length of the connection in seconds, but rather according to a coarser grid of charging units—which can be anything from a few seconds to several minutes in length, depending on the target you are calling, the time of day, etc.—the fixed solution mentioned above is not flexible enough to adapt the Short Hold timer to the changing charging unit lengths.

With release 4.6.1 you can configure your BinGO! to adapt the short hold timer dynamically depending on the actual lengths of the call charge units (*Dynamic Short Hold*).

To be able to use the Dynamic Short Hold your ISDN access must have the AOCD (advice of charge during the call[1]) feature activated.
If you are not sure whether AOCD is activated for your ISDN access, there is an easy way to verify it.
Go to the [*Monitoring and Debugging*][*ISDN Monitor*] menu of the Setup Tool while an outgoing ISDN call is active. If the *Charge* field for this call remains empty until the end of the call, no advice of charge was received during the call.

New MIB Variables

Two new MIB variables were introduced to enable Dynamic Short Hold.

- The new *ChargeInterval* variable in the ***biboPPPStatTable*** contains the length of the last charging unit. It is updated every time a new advice of charge is received.

1. Called »*Übermittlung der Tarifeinheiten* während *der Verbindung*« in Germany

The *ChargeInterval* variable is reset to 0 if no advice of charge is received for an hour.

The variable will remain at 0 if the AOCD feature is disabled, or not available from the network or PBX.

- The new *DynShortHold* variable of the ***biboPPPTable*** specifies the Dynamic Short Hold idle timer as a percentage of the current *ChargeInterval*.

  For example, if you set the *DynShortHold* variable to 75 (%), and the last measured *ChargeInterval* was 120 seconds, the idle timer will be set to 90 seconds. As soon as the *ChargeInterval* length changes, the idle timer setting will change accordingly.

Configuring Dynamic Short Hold

Static Short Hold is configured as before. Dynamic Short Hold is activated by specifying a percentage of the charge unit length (*ChargeInterval*) either in the [*WAN Partner*][*ADD*][*Advanced Settings*] menu of the Setup Tool, or in the *DynShortHold* variable of the ***biboPPPTable***.

As a default, Dynamic Short Hold is *not* active (0%).

- For *interactive connections* (e.g. telnet) you should specify a rather high Dynamic Short Hold percentage (e.g. 80-90) to avoid frequent disconnects due to short periods of inactivity.
- For *internet connections* (WWW, http, etc.) you should specify a medium to high Dynamic Short Hold percentage (e.g. 50-80) to avoid frequent disconnects due to waiting periods.
- For *data connections* (e.g. ftp) you should specify a low Dynamic Short Hold percentage (e.g. 10-40) to avoid unnecessarily waiting—and incurring charges—once a transfer is complete.

Note: 

If configured, the Static Short Hold timer will *always* take precedence over Dynamic Short Hold to avoid permanent connections.
Make sure to set the Static Short Hold to a value greater than the length of a charging unit if you want Dynamic Short Hold to have any effect.
For example, in Germany there are different maximum charging unit lengths for different tariff zones (City = 4 minutes, long distance calls = 2 minutes), so you can set the *Static* Short Hold to 245 (>4 minutes) for City connections, and to 125 (>2 minutes) for long distance calls, to avoid nullifying your Dynamic Short Hold settings.

Once the Dynamic Short Hold inactivity time is reached, the connection will be kept up until shortly before the next advice of charge is expected, thus maximizing the connection time without any additional cost.

This mechanism will not work properly for the first charging unit with a radically changed length once a new tariff zone is entered, which may result in a few inefficiently used longer charging units.

If you are using Dynamic Short Hold in connection with channel bundling, please note that the channels are released one by one, keeping open each channel until short before the next advice of charge is expected for this channel, thus maximizing the connection time without further cost.
The call will of course be disconnected immediately if either side actively closes it.

Syslog Messages

Syslog messages are created when Dynamic Short Hold disconnects a call.

```
DEBUG/PPP: dialup1: dynamic shorthold, 89 seconds after last
advice of charge

INFO/PPP: dialup1: outgoing link closed, duration 180 sec,
1950 bytes received, 1946 bytes sent, 2 charging units
```

The messages above tell you that Dynamic Short Hold disconnected the call, and supply some connection statistics.

```
DEBUG/PPP: dialup1: shorthold timeout reached

INFO/PPP: dialup1: outgoing link closed, duration 63 sec, 438 bytes
received,  434 bytes sent, 1 charging units, no AOCD
```

These messages give an example of a call disconnected by the Static Short Hold. They also tell you, that no advice of charge packets were received.

New PPP Callback Method

Overview

With release 4.6.1 there is now an additional Callback method, named **ppp_offered**. This method can be used to negotiate Callback and the number to use in the LCP—according to the LCP extensions specified in RFC 1570.

This solutions is also very flexible, because you do not need to modify the configuration at the central site each time the »sales representative« moves on to a new location—the callback number is newly transmitted at each call.

The **ppp_offered** callback has to be enabled for each partner explicitly, because it is a potential security loophole.

Explanation of Terms

In the following we will always use the term »caller« to denote the person who initially calls the central site and wishes to be called back, and the term »answerer« to denote the central site.

MIB Changes

To support the new callback method we made a few necessary changes to the MIB. The variables *biboPPPCallback* and *biboDialType* now take new values, and the ***isdnStkTable*** contains the new variable *DialOutPrefix*.

1. biboPPPCallback

biboPPPCallback now also can be set to **ppp_offered**, in addition to **enabled**, **disabled**, and **expected**, as before.

ppp_offered enables the LCP callback negotiation on the answerer side. The answerer then acknowledges the callback number sent by the caller via LCP and creates a temporary entry in the ***biboDialTable***. This entry contains the callback *Number* sent by the caller, *Type* is set to **ppp_negotiated** (see 2. below), *Direction* is set to **outgoing**, *Screening* is set to **dont_care**, *IfIndex* is set to the appropriate interface.

The temporary entry is deleted once the callback connection is closed.

When initiating the actual call back, this entry is used, unless there is an entry with the same *IfIndex*, where *Type* is **isdn** or **isdn_spv**, and *Direction* is **both** or **outgoing** (see 2. below).

Entries with *Type* **isdn** and **isdn_spv** take precedence over entries with *Type* **ppp_negotiated**.

If any or all ISDN interfaces of the answerer are connected to the ISDN via a PABX and you have to dial a prefix code for external calls these prefix codes must be configured in the variable *DialOutPrefix* in the ***isdnStkTable*** (see 3. below).

2. biboDialType

biboDialType now also can be set to **ppp_callback** and **ppp_negotiated**, in addition to **isdn**, **isdn_spv**, and **delete**, as before.

Type **ppp_negotiated** is only used by the answerer to mark the temporary entries created by an LCP callback negotiation.

Type **ppp_callback** is only used by the caller to mark entries which contain the callback number for an LCP callback negotiation.

3. isdnStkDialOutPrefix

This new variable in the *isdnStkTable* must be used to configure prefix codes if any or all of the answerer's ISDN interfaces are connected to the ISDN via a PABX. In many cases this is simply the digit »0«.

This prefix is then dialled before the callback number supplied by the caller.

At the moment this variable is exclusively used for callbacks initiated via an LCP callback negotiation.

4. Microsoft Extension CBCP (CallBack Control Protocol)

The LCP callback negotiation as specified in RFC 1570 has a few inherent disadvantages, e.g. when using inband authentication the callback number supplied by the caller is acknowledged by the answerer *before* identifying the caller.

Microsoft proposed CBCP as an internet draft (»Proposal for CallBack Control Protocol (CBCP)«, July 19, 1994) to remedy some of these disadvantages.

Microsoft software uses CBCP for connections made via Dial-Up Networking as well as via Remote Access Service (RAS) Server / Client.

As a default LCP callback negotiation as specified in RFC 1570 is used for connections from BRICK to BRICK. When communicating with Windows systems CBCP is also accepted.

5.a Configuring Callback on the BinGO!

Create a WAN partner entry with Authentication set to CHAP (in Setup Tool set the *PPP Authentication Protocol* to **CHAP** in the [*WAN Partner*][*ADD*] menu), and *Callback* set to **ppp_offered** (in Setup Tool set the *Callback* to **offered** in the [*WAN Partner*][*ADD*][*Advanced Settings*] menu).

Make sure that *Identify by Calling Number* is set to **no** for modem call-ins. Fill in the other fields as appropriate for this partner.

5.b Configuring Callback under Windows 95

Callback is not explicitly configured under Windows 95. Use your normal Dial-Up Networking settings.

During connection setup the BinGO! offers Callback, the PC then opens a dialogue box where the user can enter the number

to use for this callback. Once this has happened the connection is cleared, and the BinGO! initiates the callback.

Note: The initial connection will remain open until the user has entered a callback number.

5.c Configuring Callback under Windows NT

Under Windows NT[1] go to Dial-Up Networking and click on the »More« button.

Then select »*Edit entry and modem properties…*«.

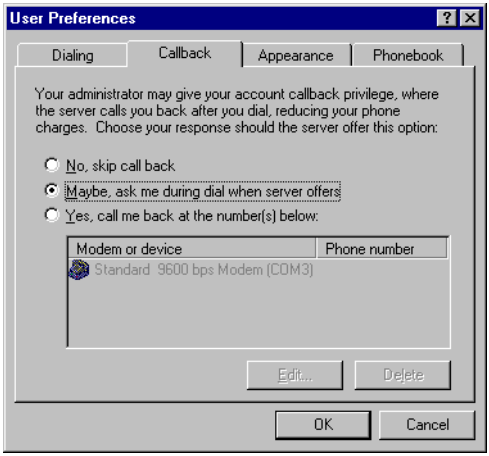
Configure the basic settings for this entry, on the Server page select the Dial-up server type »*PPP: Windows NT, Windows 95 Plus, Internet*«, select TCP / IP as a network protocol, and *Enable PPP LCP extensions.*

On the Security page select »Accept only encrypted authentication«.

Confirm your settings with »OK«.

Finally, click on »More« once again and select »*User preferences…*«. There you can configure callback.

1. This example is for Windows NT Client 4.0, SVP 3, the dialogues may look slightly different on your system.



Choose »Maybe«. The PC will then ask you to specify the call-back number, as described above for Windows 95.

Note:

With release 4.6.1 it is not yet possible to use the BinGO! as a caller, when the answerer is a Windows NT RAS server. The BinGO! can then only act as the answerer.