



SYSTEM ADMINISTRATION

September 2000





SYSTEM ADMINISTRATION

A	REFERENCE	5
1	System Administration on the BinTec router	6
1.1	System Logging on the BinTec router	6
1.1.1	Accounting Messages and System Messages	8
1.2	Gathering Accounting Information	13
1.2.1	ISDN Accounting Information	13
1.2.2	IP Accounting Information	17
1.3	Credits Based Accounting System	19
1.3.1	ISDN Channel Reservation	20
1.4	Logging with Remote LogHosts	23
1.5	Remote SNMP Administration	26
1.5.1	Traps	27
1.6	Keepalive Monitoring	29
1.7	Windows Activity Monitor	36
1.8	Web Based Monitoring	39
1.9	User Accounts	46
1.10	Other Passwords	47
1.11	System Software Updates	48
1.11.1	What's Needed	48
1.11.2	Performing a System Software Update	48
1.12	BOOT Options on the BinTec router	50
1.12.1	The BOOTmonitor	50

1.12.2	Booting via BootP	54
1.12.3	BootP Relay Agent	55
1.13	Other System Administration Tasks	56
1.13.1	Setting Up a BootP Server	56
1.13.2	Setting up a TFTP Server	57
1.13.3	Setting Up a syslog Daemon	59
1.13.4	Setting up a Time Server	61
1.14	The Modem Function Module	63
1.14.1	V.90/K56flex Modem Function Module	63
1.14.2	Introduction	64
1.14.3	Hardware	66
1.14.4	Software	67
1.14.5	WAN Partner / Outgoing Calls	74
1.14.6	Example Configuration	77
1.14.7	Tracing a Modem Connection	84

REFERENCE

1 System Administration on the Bin-Tec router

1.1 System Logging on the BinTec router

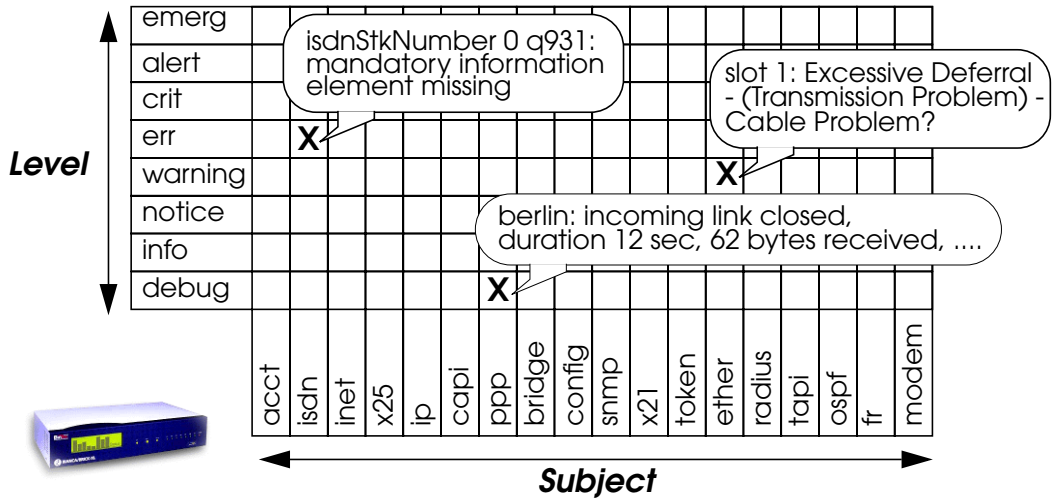
During normal operation various messages may occasionally be generated on the BinTec router by its various subsystems (ISDN, PPP, X.25, MODEM, etc.). These messages, called syslog messages, are generated in response to error conditions or other events that may occur while the system is running.

A syslog message is a text string consisting of four pieces of information relating to the event that occurred. Syslog messages are stored locally in the BinTec router's *biboAdmSyslogTable*. A limited number of messages are saved here (defined by the value of the *biboAdmSyslogMaxEntries* object, default is 20); each time the system reboots existing messages are lost.

The *biboAdmSyslogTable* consists of the following fields.

<i>TimeStamp</i>	A date string of the format: MM/DD/YY HH:MM:SS that identifies the date and time the message was generated.
<i>Level</i>	The severity of the event; i.e., the higher the level the more important the message is considered (see below).
<i>Message</i>	The actual text of the message. The text attempts to describe the circumstances relating to the event.

Subject The internal software subsystem that generated the message.



Recent system messages can be displayed from the SNMP shell at any time by entering **message** at the shell prompt.

1.1.1 Accounting Messages and System Messages

Syslog messages fall into two categories; Accounting messages and System messages. Accounting messages are generated by the **acct** subsystem. (See the *Subject* field of the *biboAdmSyslogTable* above.) System messages are generated by any of the other BinTec router subsystems which, depending on the current license(s) installed on the system, may include:

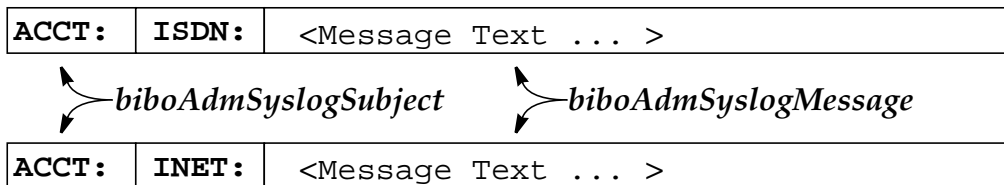
```

isdn   inet   x25   ipx   capi  ppp
bridge config snmp  x21   token ether
radius tapi  ospf  fr    modem

```

Accounting Messages

Accounting messages are used to report accounting information relating to either an ISDN connection or an IP session that was closed/routed over the BinTec router. Accounting messages are identified (in the *biboAdmSyslogTable* or in a remote file on a LogHost where syslog messages are being sent) by an initial **ACCT:** tag in the text of the message. For ISDN messages, the **ISDN:** tag immediately follows; for IP accounting messages **INET:** follows.



ISDN Accounting Messages

An ISDN accounting message contains information regarding an ISDN call that was either placed or received

by the BinTec router. Details for both successful and unsuccessful ISDN outgoing calls are reported here.

Note



If a B-channel is used to its full capacity for at least three days, ISDN Accounting information can overload resulting in the subsequent sending of erroneous accounting messages.

The content and format of ISDN accounting messages vary according to the special formatting tags contained in the *isdnAccountingTemplate*. A list of possible format tags that can be used in the accounting template and their meanings are shown below.

Format Tag	Meaning
%S	Date the connection opened; in DD.MM.YY format.
%s	Time the connection was established: in HH:MM:SS format
%R	Date the connection closed; in DD.MM.YY format.
%r	Time the connection was closed: in HH:MM:SS format.
%d	The duration of the connection in seconds.
%y	Total number of bytes received over the connection.
%Y	Total number of bytes sent over the connection.
%g	Total packets received over the connection
%G	Total packets sent over the connection.
%c	Total number of charging units (value) incurred for the connection.
%C	Total number of charging units (string) incurred for the connection.

Format Tag	Meaning
%n	The call's direction; either incoming or outgoing.
%Z	The local address (Calling or Called party's number, see %n).
%z	The local subaddress (Calling or Called party's number, see %n).
%T	The remote address (Calling or Called party's number %n).
%t	The remote subaddress (Calling or Called party's number %n).
%i	Service indicator and additional information for the call.
%b	Bearer capability for the call.
%l	Low layer capability for the call.
%h	High layer capability for the call.
%u	DSS1 error cause, if applicable.
%U	1TR6 error cause, if applicable.
%L	Local (BinTec router internal) error cause.
%F	Call reference (BinTec router internal).
%I	Information about the BinTec router subsystem the call was given to.

The default accounting template setting contains the following tags:

**%S,%s,%r,%d,%y,%Y,%g,%G,%C,%n,%Z,%T,%i,
%u,%L**

This template produces accounting messages similar to the following.

```

IS-
DN:18.08.1997,13:53:19,13:53:34,12,1096,1875,33,33,1Units,O,2,0030399
88452,7/0,9F,0

```

Changing the ISDN Accounting Template

The accounting template can be changed to meet your particular needs. As shown [above](#) the comma character is used as the default delimiter, separating each data field. However, since the *isdnAccountingTemplate* is a quoted string arbitrary words and characters may be added as needed.

This may be useful for sites forwarding accounting messages to remote UNIX loghosts and performing post-processing (via `grep` or other shell scripts). Setting the accounting template to the value:

```

"%S## LinkUp@%s-Down@%r (Called %n->to %T)
                                %c charging units"

```

would result in less informative, more readable messages similar to:

```

18.08.1997## LinkUp@17:36:08-Down@17:36:10(Called Out->to 254)
                                0 charging units
18.08.1997## LinkUp@17:36:08-Down@17:36:10(Called in->to 187)
                                7 charging units
18.08.1997## LinkUp@17:36:08-Down@17:36:10(Called Out->to 794)
                                5 charging units
18.08.1997## LinkUp@17:36:08-Down@17:36:10(Called Out->to 234)
                                4 charging units

```

IP Accounting Messages

IP accounting messages contain information for a specific IP session that was routed over the BinTec router. In

contrast to ISDN accounting messages, IP accounting messages have a fixed format and can't be changed. A sample IP accounting message showing the respective fields is shown below.

14.08.1997 10:57:06 124 6 10.5.5.5:1036/1000 -> 10.2.2.2:21/10002 1 71 1 144

↑ Date this IP session was established
 ↑ Time this IP session was established
 ↑ Session duration
 ↑ Protocol Identifier
 ↑ Source IP address
 ↑ Source Port
 ↑ Source Interface
 ↑ Destination IP Addr
 ↑ Destination Port
 ↑ Destination Interface
 ↑ Packets sent
 ↑ Bytes sent
 ↑ Packets received
 ↑ Bytes received

IP accounting messages are only generated for IP sessions routed over IP interfaces for which accounting has been enabled. This is done by setting the respective *ipExtIfAccounting* variable in the *ipExtIfTable* is set to **on**.

Once accounting for an interface is turned on, active IP sessions routed over the interface appear in the *ipSessionTable*. Once a session closes, either by disconnection or timeout, an accounting message is generated and is written to the *biboAdmSyslogTable*.

System Messages

System messages are generated by BinTec router system software subsystems in response to certain errors or events. Recall that all syslog messages include a BinTec router subsystem tag at the beginning of the message text. System messages are identified by any subsystem tag other than the **ACCT:** tag.

The most common system messages are shown in [Appendix E](#).

1.2 Gathering Accounting Information

Accounting information relating to active or closed ISDN connections or IP sessions on the BinTec router can be queried locally on the BinTec router via various system tables or logged to remote hosts using the syslog protocol.

1.2.1 ISDN Accounting Information

ISDN accounting messages contain information about ISDN calls that was either placed or received by the BinTec router.

Tracking Current ISDN Connections

Statistics for current ISDN calls are stored in the *isdnCallTable*. As long as the call is active, the corresponding fields in this table are updated. Once an ISDN call is closed, or disconnected, the *isdnCallTable* entry is removed and a new entry is created (using the data from the *isdnCallTable* entry) in the *isdnCallHistoryTable*.

To show how these table entries are created/removed, we'll establish a loopbacked ISDN connection to our BinTec router using our own ISDN telephone number (143) in the example below. This assumes that incoming call dispatching has been configured allowing calls to 143 to be given to the login service.

Using the **isdnlogin** program we place the call and login as **admin**.

```
mybrick: system> isdnlogin 143
```

```
Trying...
```

```
Establishing B-channel...
```

```
Connected to 143
```

```
Connected to BIANCA/BRICK-XS, mybrick, Germany
```

```
Welcome to BIANCA/BRICK-XS version V.4.5 Rev.3 from 97/08/01 00:00:00  
systemname is mybrick, location Germany
```

```
Login: admin
```

```
Password:
```

```
mybrick: >
```

Then we simply display the *isdnCallTable* to see the details of active ISDN connections.

```

mybrick: > isdnCallTable
inx StkNumber(*ro)      Type(*ro)      Reference(*ro)
  Age(ro)               State(rw)      IsdnIfIndex(ro)
  Channel(ro)           DspItem(ro)    RemoteNumber(ro)
  RemoteSubaddress(ro) LocalNumber(ro) LocalSubaddress(ro)
  ServiceIndicator(ro) AddInfo(ro)     BC(ro)
  LLC(ro)               HLC(ro)        Charge(ro)
  ReceivedPackets(ro)  ReceivedOctets(ro) ReceivedErrors(ro)
  TransmitPackets(ro)  TransmitOctets(ro) TransmitErrors(ro)
  ChargeInfo(ro)       Screening(ro)   Info(ro)

00 0                    outgoing      4
  0 00:26:30.00        active        2000
  1                    login        "143"

  data_transfer        0            88:90
                               0
  553                  2754        0
  542                  8357        0
                               undefined    "isdnlogin"

01 0                    incoming      2
  0 00:26:30.00        active        2000
  2                    eaz3
                               "3"
  data_transfer        0            88:90
                               0
  558                  2834        0
  572                  9183        0
                               undefined    "isdnlogind"

mybrick:isdnCallTable> exit

```

Since we placed a loopbacked call by calling our own ISDN number a separate entry is present for both the incoming and the outgoing call.

The *Type* field (shown above) identifies the direction of the call. Details of the ISDN call are contained in the respective fields most of which are self explanatory. For information regarding the meanings of specific fields refer to the MIB reference contained on the Companion CD.

We can terminate the ISDN connection by ending the `isdnlogin` session started previously. The `isdnCallTable` entry is dismissed and a new `isdnCallHistoryTable` entry is created as shown below. Again, since an incoming and an outgoing call was registered, two entries are added to the `isdnCallHistoryTable`.

```
mybrick: > isdnCallHistoryTable
inx StkNumber(*ro)      Type(*ro)              Time(ro)
Duration(ro)           IsdnIfIndex(ro)       Channel(ro)
Dsptem(ro)             RemoteNumber(ro)      RemoteSubaddress(ro)
LocalNumber(ro)        LocalSubaddress(ro)   ServiceIndicator(ro)
AddInfo(ro)            BC(ro)                LLC(ro)
HLC(ro)                Charge(ro)            DSS1Cause(ro)
TR6Cause(ro)           LocalCause(ro)        ChargeInfo(ro)
Screening(ro)          Info(ro)

00 0                    incoming               08/19/97 13:28:25
39                      2000                   2
eaz3
"3"
0                        88:90                  data_transfer
0x80                     0                       0x9f
undefined                "isdnlogind"

01 0                    outgoing               i08/19/97 13:28:25
39                      2000                   1
login                    "143"
0                        88:90                  data_transfer
0x80                     0                       0x9f
undefined                "isdnlogin"
```

mybrick: isdnCallHistoryTable>

Note:

The number of entries in the `isdnCallHistoryTable` is limited to the value set in the `isdnHistoryMaxEntries` object. By default information regarding the last 20 ISDN calls are saved with older entries being dismissed as newer entries are added.

Most fields shown above are self explanatory. For meanings of the *DSS1Cause*, *1TR6Cause*, and *LocalCause* fields, refer to [Appendix C](#).

For descriptions regarding the meanings of individual fields in the `isdnCallHistoryTable` see the MIB Reference contained on the Companion CD.

Logging ISDN Accounting Information to LogHosts

ISDN accounting messages can be forwarded to remote hosts for storage or post processing. This is done by configuring the remote host as a LogHost on the BinTec router in the *biboAdmLogHostTable*. LogHosts may include PCs running *DIME Tools Syslog Daemon* program (see: [BRICK-ware for Windows](#)) or a UNIX workstation where the syslog daemon is appropriately configured (see: [Setting Up a syslog Daemon](#)).

To configure the LogHost on the BinTec router refer to the section on: [Logging with Remote LogHosts](#).

Note:



When configuring LogHosts for accounting information ALL accounting information (both ISDN and IP accounting messages) will be sent to this host.

1.2.2 IP Accounting Information

IP accounting messages contain information about a specific IP session routed over the BinTec router. Recall that IP accounting messages are only generated for IP sessions that are routed over interfaces for which IP accounting has been enabled in the *ipExtIfTable*.

Tracking Active IP Sessions

Statistics for active IP sessions routed over BinTec router interfaces (again, interfaces for which IP accounting is enabled) can be seen in the *ipSessionTable*. Once an IP session closes this entry is removed and a IP accounting message is generated and saved to the *biboAdmSyslogTable*.

The SNMP session shown below displays the respective table entries that might be created for an FTP session between a host on the BinTec router's LAN (*ifIndex* = 1000 IP Address = 192.168.2.2) and a remote host via a dial-up link (*ifIndex* = 10002 IP Address = 10.5.5.5).

```
mybrick: > ipSessionTable
```

inx	SrcAddr(*ro) OutPkts(ro) Protocol(*ro) DstIfIndex(ro)	SrcPort(*ro) OutOctets(ro) Age(ro)	DstAddr(*ro) InPkts(ro) Idle(ro)	DstPort(*ro) InOctets(ro) SrcIfIndex(ro)
00	192.168.2.2	1224	10.5.5.5	21
	45	1860	28	1570
	tcp	0 00:00:10.00	0 00:00:00.00	1000
	10002			

```
mybrick: ipSessionTable>
```

Once the session closes an entry is made to the *biboAdmSyslogTable* and if applicable, a message is sent to the configured LogHost(s). When displaying the *biboAdmSyslog-*

Table only the first few characters of the message text is displayed. To see the full text enter the **message** command.

```
mybrick: > biboAdmSyslogTable
```

inx	TimeStamp(*ro)	Level(*ro)	Message(ro)	Subject(ro)
00	01/01/70 0:00:09	err	"TIMED: no respon	inet
01	08/19/97 19:07:35	info	"INET: 19.08.1997	acct

```
mybrick: ipSessionTable>message
```

```
00 "TIMED: no response"
01 "INET: 19.08.1997 18:55:25 709 6 192.168.2.2:1224/1000 -> 10.5.5.5:21/10002 61 2506 41 2380"
```

```
mybrick: ipSessionTable>
```

Logging IP Session Information to LogHosts

IP accounting messages can be forwarded to remote hosts configured to accept syslog messages. Such hosts may include PCs running the included *DIME Tools Syslog Daemon* program (see: [BRICKware for Windows](#)) or a UNIX workstation where the syslog daemon is appropriately configured (see: [Setting Up a syslog Daemon](#)).

To configure the LogHost on the BinTec router refer to the section on: [Logging with Remote LogHosts](#).

Note:



When configuring LogHosts on the BinTec router for accounting information ALL accounting information (both ISDN and IP accounting messages) will be sent to this host.

1.3 Credits Based Accounting System

With dial-up WAN connections it may occur that charges rise because of configuration errors. The Credits Based Ac-

counting System gives BinTec router administrators the ability to control charges. It allows the BinTec router 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 BinTec router 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 *isdnCreditsTable* controls this feature, it is described in the current MIB Reference <http://www.bintec.de/download/brick/doku/mibref/index.html>.

The Credits Based Accounting System can also be configured via Setup Tool: in the main menu over **ISDN** to manage and activate the system; and over **Monitoring and Debugging** to monitor the incoming and outgoing connections and accounted charges.

1.3.1 ISDN Channel Reservation

In order to give you even more control over the number and direction of your calls, the *isdnCreditsTable* allows you to set the maximum number of incoming calls, outgoing calls, as well as the total number of calls being made at the current moment in time.

- Example:
This means that if it is very important for you that at least half of your PRI B-channels (30 in total) remain open for incoming calls, you can set the *MaxCurrentOutCon* variable (the maximum number of outgoing calls), to 15, this would mean that at least 15 channels would be reserved for incoming calls.

Configuration

The relevant variables in the *isdnCreditsTable* can be configured in the MIB using the SNMP shell:

Variable	Meaning
<i>MaxCurrentInCon</i>	This variable allows you to set the maximum number of current incoming connections
<i>MaxCurrentOutCon</i>	This variable allows you to set the maximum number of current outgoing connections.
<i>MaxCurrentCon</i>	This variable allows you to set the maximum number of incoming as well as outgoing calls currently being made.

MaxCurrentInCon and *MaxCurrentOutCon* can be configured in Setup Tool in the menu

ISDN → CREDITS → EDIT →

BinTec router Setup Tool (ISDN)(CREDITS)(EDIT): Configure ppp Credits	BinTec Communications AG MyRouter
Surveillance	on
Measure Time (sec)	6400
Maximum Number of Incoming Connections	off
Maximum Number of Outgoing Connections	on
Maximum Charge	100 off
Maximum Time for Incoming Connections (sec)	on 28800
Maximum Time for Outgoing Connections (sec)	on 28800
Maximum Number of Current Incoming Connections	on 1
Maximum Number of Current Outgoing Connections	on 1
SAVE	CANCEL
Enter integer range 0..2147483647	

- Activate *Maximum Number of Incoming/Outgoing Connections* by pressing the Spacebar and changing off to on.
- Enter the number of B-channels you want to reserve for that direction.
- Press *SAVE*.

In the above example, the two new variables at the bottom of the table are both set to one. This means that it is impossible to make more than one incoming or outgoing call at any one given time.

Surveillance

It is also possible to observe the number of connections currently being made by going to

Monitoring and Debugging → ISDN Credits → Subsystem

BinTec router Setup Tool (Monitor) (CREDITS) (STAT): Monitor ppp Credits		BinTec Communications AG MyRouter	
	Total	Maximum	%reached
Time till end of measure interval (sec)	84400	86400	2
Number of Incoming Connections	1		1
Number of Outgoing Connections	1		1
Time of Incoming Connections	734		
Time of Outgoing Connections	244		
Charge	0		
Number of Current Incoming Connections	22	22	
Number of Current Outgoing Connections	6	12	
Number of Current Connections	28	28	
Exit			

In this case, the new variables show that twenty-two incoming calls are being made out of a possible total of twenty-two, only six of the permitted twelve outgoing calls are being made, however, because the maximum number of current connections is restricted to twenty-eight.

1.4 Logging with Remote LogHosts

LogHosts are configured on the BinTec router in the *bi-boAdmLogHostTable*. This table consists of four fields that define the following attributes for the LogHost.

<i>Addr</i>	The IP address of the host to send the syslog message to.
<i>Level</i>	The level of syslog messages to send to this host. This is a minimum level; setting this object to level X sends all messages with levels $\geq X$ (See: System Logging on the BinTec router).
<i>Facility</i>	This is the syslog facility on the LogHost the BinTec router sends the message to.
<i>Type</i>	This is only required for UNIX LogHosts. The type (either system , accounting , or all) of syslog messages to send to this host. System and accounting messages are described here , all include both types.

LogHosts configured on the BinTec router must be configured to accept messages via the syslog protocol. For PCs the *DIME Tools Syslog Daemon* can be used. For UNIX workstations, the `syslogd` must be properly configured and running (see: [Setting Up a syslog Daemon](#)).

The BinTec router always uses the UDP port 514 for sending syslog messages.

A simple LogHost setup involving one one remote host is shown below. In this example accounting messages, both ISDN and IP, and system messages with levels \geq **err** are sent to this host.

Note:

Because of cost considerations it is generally not a good idea to configure LogHosts that are only accessible via ISDN DialUp links.

Since we want to keep our accounting and system warning messages in separate files on the remote LogHost we need to make two entries in the *biboAdmLogHostTable*.

```
mybrick: > biboAdmLogHostTable

inx Addr(*rw)          Level(-rw)          Facility(rw)          Type(rw)

mybrick: biboAdmLogHostTable> Addr=192.168.5.99 Level=info Facility=local0 Type=acct
mybrick: biboAdmLogHostTable> Addr=192.168.5.99 Level=err Facility=local1 Type=system

mybrick: biboAdmLogHostTable> biboAdmLogHostTable

inx Addr(*rw)          Level(-rw)          Facility(rw)          Type(rw)

00 192.168.5.99        info                local0                acct
01 192.168.5.99        err                 local1                system
mybrick: biboAdmLogHostTable>
```

Note:

Accounting messages are generated at the Level=info. If you configure a log host for accounting messages (Type=acct) and specify a level higher than info no messages will be sent to the LogHost.

Assuming our UNIX LogHost was configured to accept these syslog messages via the **local0** and **local1** facilities and save the information to the **/var/adm/mybrick.acct** and **/var/adm/mybrick.system** files respectively, we might see the following information accumulate there.

/var/adm/mybrick.acct

```
Aug 14 11:19:26 mybrick ACCT: INET: 14.08.1997 11:18:46 1 6
                10.2.2.6:2855/4000->10.4.5.8:25/10002 30 16 1000
Aug 14 11:24:48 mybrick ACCT: ISDN: 14.08.1997,11:24:08,11:24:22,
                12, 1185, 2715,37,37,1 Units,O,2,7834,7/0,9F,0
```

/var/adm/mybrick.system

```
Aug 14 11:23:54 mybrick ISDN: isdnStkNumber 0 q931:
                information element missing
```

The initial date and time strings at the beginning of the message are set by the local host (or PC). They reflect the date and time the message was received and may not correspond to the actual time of the system event.

1.5 Remote SNMP Administration

Object Identifiers (OIDs)

All OIDs of all MIB variables have the same structural form.

.1.3.6.x.x.x.x .y.y.y .i

.1.3.6.x.x.x.x : is the OID of the variable according to the MIB description file

.y.y.y : is the specific OID part for the unambiguous identification of a variable in several rows of a dynamic table (non-existent in static tables). It consists of the contents of all index variables (*variables), which are mostly unambiguous by row.

For tables where this is not the case (e.g. ipRouteTable), the following index is required for purposes of clarity.

.i : is a continuing index (always 0 for static tables) not the same as the 'inx' on the Command Line.

The Raw-Mode (numerical form) command `x` toggles Raw-Mode on and off. After entering the command, the shell reports which mode it is entering. By default Raw-Mode is off from the SNMP shell.

1.5.1 Traps

Standard and Enterprise-Specific Traps

To report asynchronous events to a management station (trap host) the BinTec router can send traps. Asynchronous events means e.g. the change of a MIB variable, which may require attention. Traps are differentiated into Standard Traps and Enterprise-Specific Traps.

Standard Traps report the events “coldStart, warmStart, linkDown, linkUp and authenticationFailure” and are sent by default when a trap host is defined or trap broadcasting is turned on.

coldStart	Reboot of the BinTec router
warmStart	Reboot of the BinTec router
linkDown	Disconnection of a link. Change of the variable ifOperStatus to the value down or dormant (hardware and software interfaces).
linkUp	Establishment of a connection. Change of the variable ifOperStatus to the value up (hardware and software interfaces).
authenticationFailure	An SNMP authentication failure, i.e. SNMP request with wrong password.

Enterprise-Specific Traps can be defined by the user. To define a trap object the user must assign a MIB variable (object identifier in dot format or string) to the variable **biboATrpObj** in the **biboAdmUserTrapTable**.

Only certain variables, which could contain important changes, can be trapped. Counters can not be trapped.

Traps can either be broadcasted to the local LAN or be sent to a defined trap host. Trap hosts can be configured in the **biboAdmTrapHostTable**.

Broadcasting traps into the LAN can be configured with the variable *biboAdmTrapBrdCast* in the *adminTable*, where also the *TrapPort* (default: 162) and the *TrapCommunity* (default: "snmp-Trap") can be adjusted.

The following two examples explain the structure of trap packets, which are ASN1 coded:

Standard Trap:

Trap Item	Meaning
"snmp-Trap"	trap community
.1.3.6.1.4.1.272	enterprise OID (=.iso.org.dod.internet.private.enterprise.bintec)
192.1.2.3	IP address
linkUp	trap type (coldStart, warmStart, linkDown, linkUp, authenticationFailure)
0	no meaning
0:33:58	time stamp
"BIANCA/BRICK-XL"	system description
"brick"	system name
ifoperstatus.10001.4 = up	interface state

Enterprise-Specific Trap:

Trap Item	Meaning
"snmp-Trap"	trap community
.1.3.6.1.4.1.272	enterprise OID (=.iso.org.dod.internet.private.enterprise.bintec)

Trap Item	Meaning
192.1.2.3	IP address
enterprise specific	trap type
row identifier (integer)	table number (=table number * 1000 + row number)
0:33:58	time stamp
"BIANCA/BRICK-XL"	system description
"brick"	system name
isdnchState.2000.1.1 = connected	trap variable

1.6 Keepalive Monitoring

Keepalive Monitoring is a feature that prevents unnecessary connections being made by a central server over the WAN to the router of a LAN. If that server regularly tries to transfer data to the router and its hosts, it may well happen that there are no computers turned on in the LAN and the calls made to the router as well as the costs incurred, of course, are in vain.

The solution to this problem is a monitoring system performed by the router which checks at regular definable intervals whether computers in the router's LAN can be reached or not. By means of a new table, the *ipHostsAliveTable*, the state of specified IP addresses can be monitored by the router. The router tests the accessibility of the computers by pinging them. If, after three attempts, no computer in the LAN is reachable, PPP connections are deactivated by changing the state of the corresponding interface to *down*. If, on the other hand, at least one computer in the LAN responds to the ping, the state of the IP address is

set to *alive* and the status of the corresponding interface (*ifAdminStatus*) is set to *up* and PPP connections can be made.

The reachability of the IP address is thus not initiated by the client PC itself. Therefore, the connection of router to client is not necessarily made immediately after the PC is turned on. The connection is made only after the router itself has established the reachability of the client, a process which is dependent on the monitoring times set.

WAN Partner Configuration

Essentially, Keepalive Monitoring oversees the accessibility of computers in a LAN and controls the states of the interfaces of WAN partners accordingly. An incoming call from a WAN partner can respond to the state of the interfaces or not, depending on how that WAN partner is configured. Let's look at some of the implications for the effectiveness of Keepalive Monitoring of different WAN partner configurations.

1. WAN partners identifiable by CLID
If the WAN partner is configured and identifiable by CLID, the call is accepted if the state of that WAN partner's interface (*IfAdminStatus*) is *up*; the call is refused if the state of the interface is *down*. Thus, thanks to Keepalive Monitoring, no unnecessary connections are established.
 - In the event of an inband call (non-CLID)
 - If just the one WAN interface is configured with non-CLID, the functionality of Keepalive is guaranteed. When the configured inband WAN partner calls, if the interface has been set to *up*, the call will be correctly taken as there is only one non-CLID interface to match the inband (non-CLID) call.
 - If two or more WAN interfaces are configured with non-CLID and these interfaces are *up*, an inband

call will be established although an unnecessary connection may be made. In such a case, the purpose of Keepalive Monitoring, i.e. to prevent unnecessary connections, is not served.

2. If ALL interfaces are down, an inband call is not established.
3. Keepalive Monitoring has no effect for calls coming from partners configured on a RADIUS server. In this case, there is no static interface defined for that partner on the BinTec router that can be influenced by the monitoring feature.

The meanings of the different variables from the *ipHostsAliveTable*:

Variable	Meaning
ipHostsAliveGroup	This is a group of IP addresses that can be monitored. The number of these groups is restricted to ten, the range of values spans from 0 to 9 (0 being the first group, 9 the last). IP addresses in the same group are combined with OR, which means that as long as at least one IP address is reachable in the group, the state of the group is set to <i>alive</i> . The state of the entire group is only set to <i>down</i> when none of the IP addresses in the group is reachable.
ipHostsAliveIPAddress	An IP address that is to be monitored or pinged by the router. The number of IP addresses per group is also restricted to 10, which means that, in total, up to 100 IP addresses can be monitored.
ipHostsAliveState	Here the state of the monitored IP addresses is maintained. The possible values are either <i>alive</i> or <i>down</i> . <i>Alive</i> if the IP address is reachable; <i>down</i> if the IP address is not reachable.
<i>ipHostsAliveInterval</i>	The interval calculated in seconds between which the IP addresses are pinged. The range of values spans from 1 to 65536. (The default value is set at 300 seconds = 5 minutes.) Each IP address is arranged in a group and is given an interval time. A monitoring time is then calculated for each group where the shortest interval of all is used. Then a global monitoring time is calculated using all the group values, again the shortest value is used. The global monitoring time is used to start the program, after booting it could take 2 minutes (if that is the shortest group interval) for the first ping to take place, during which the state of the IP addresses is set to 0. Hosts are then pinged according to the group monitoring time. If there is no reply, the state of the IP address can be reset after three pings. If there is a reply, the state of the IP address changes immediately.

Variable	Meaning
<i>ipHostsAliveDownAction</i>	This variable defines the effect on the defined interfaces if <i>ipHostsAliveState</i> changes to the <i>down</i> state. The possible values are <i>up</i> , <i>down</i> or <i>delete</i> (<i>delete</i> does not affect the <i>ifAdminStatus</i> , but merely deletes the entries of that line in the MIB table), the default is <i>down</i> . The <i>ifAdminStatus</i> of the interface(s) is then changed according to the value in <i>DownAction</i> (see below).
<i>ipHostsAliveFirstIfIndex</i>	This is the first interface number in the index list that is to be affected by the execution of <i>DownAction</i> .
<i>ipHostsAliveRange</i>	This indicates the possible range of numbers of interfaces that can be affected by the execution of <i>DownAction</i> . So if you set <i>FirstIfIndex</i> to 10001 and <i>Range</i> to 0, only the interface 10001 is affected and none other. On the other hand, if, in the same case, you set <i>Range</i> to the default number 4999, all the interfaces with a number between 10001 and 15000 are affected by <i>DownAction</i> .

Example of an entry in the new IP table (***ipHostsAliveTable***):

inx	Group(rw) DownAction(rw)	IPAddress(*rw) FirstIfIndex(ro)	State(ro) Range(rw)	Interval(rw)
00	5 down	192.168.98.129 10003	alive 0	180

This means that the IP address ***192.168.98.129*** belongs to group 5. The interval between pings is set to 3 minutes. If this IP address is no longer reachable (and, if there are any, all other IP addresses in group 5 are also unreachable), the ***ifAdminStatus*** of the interface 10003 is set to the *down* state.

Possible states

Depending on the state of the group and what is specified in *DownAction*, the *ifAdminStatus* for the defined interfaces receive the following values:

<i>Group State</i>	<i>AND</i>	<i>DownAction</i>	<i>=></i>	<i>ifAdminStatus</i>
alive (reachable)	+	down	=>	<i>ifAdminStatus_up</i>
alive (reachable)	+	up	=>	<i>ifAdminStatus_down</i>
down (not reachable)	+	down	=>	<i>ifAdminStatus_down</i>
down (not reachable)	+	up	=>	<i>ifAdminStatus_up</i>

As the table shows, when the group state is *alive* or reachable, the *ifAdminStatus* is set to the contrary of what is specified in *DownAction*. Conversely, when the group state is set to *down*, the *ifAdminStatus* conforms with the specification under *DownAction*.

One IP address in two groups?

If the same IP address is in two or more different groups with different monitoring times (can occur if the same IP address is allocated to different interfaces), it can happen that the group states of the one IP address are different, as well as the states of the respective interfaces.

For example: let's assume the same IP address appears in two groups. In the first of these groups, the monitoring time is set to 60 seconds, in the second to 300 seconds. If, in both cases, the IP address is not reachable, the state of the first group will be set to down after 3 x 60 seconds; the second, after 3 x 300 seconds. There could thus be an interim period

when the state of group 1 is set to down and the state of group 2 is set to up. This would mean in turn that the interface of the IP address in group 1 is set to down in the *ifAdminStatus*, the interface of the same IP address in group 2 is set to up.

However, once the IP address is up again, the state of the IP address changes in both groups immediately after the reply to the ping from the first group.

SysLog messages

The following is a list of relevant SysLog messages:

biboAdmSyslogMessage	Log Level
alivepacket sent to xxx.xxx.xxx.xxx (a packet has been sent to the following IP address xxx.xxx.xxx.xxx)	debug
xxx.xxx.xxx.xxx is alive (a packet has been received from the IP address xxx.xxx.xxx.xxx)	debug
no answer from xxx.xxx.xxx.xxx (after three ping attempts, no packet was received from the IP address xxx.xxx.xxx.xxx)	debug
only 10 entries per group are allowed (the user wanted to create more than ten entries in the one group. The entry last attempted is not included in the MIB table)	err
cannot send alivepacket to xxx.xxx.xxx.xxx (a packet could not be sent to the IP address xxx.xxx.xxx.xxx)	err
interface xxxxx set (up down) (the interface numbered xxxxx was set to the up or down state)	info

Changing the state of the interface over the SNMP shell

If the state of an interface is changed by the user in the SNMP shell, the program does not register the change. The following pattern of behaviour is possible:

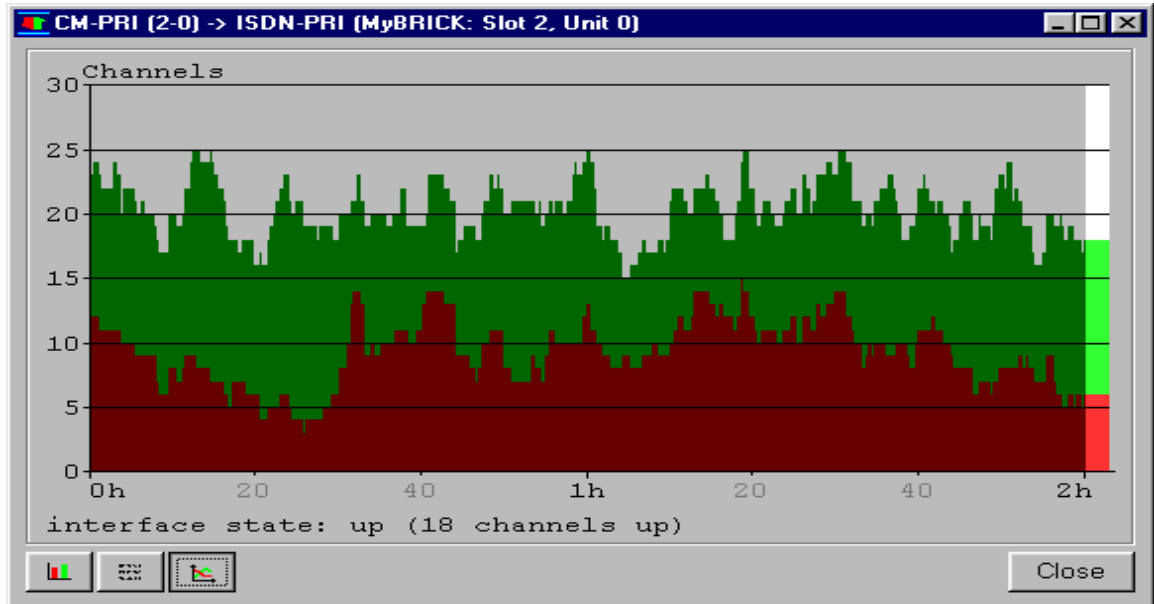
If the state of the IP address is down and *DownAction* is also set to down, but the user manually sets the *ifAdminStatus* to up in the SNMP shell, the interface status remains up even if, subsequent to the next three pings, the program has established that the state of the IP address is down. The interface status will remain up as long as the IP address remains down.

However, once the IP address responds to a ping (i.e. the IP address is reachable and its state is thus set to alive), the program once again has control over the interfaces. Although this means that nothing changes in this case, i.e. the *ifAdminStatus* remains up, the next time this IP address can not be reached and its state is set to down, the *ifAdminStatus* of the corresponding interface is also set to down.

1.7 Windows Activity Monitor

Why the Activity Monitor?

With the Activity Monitor, Windows users can monitor the activities of a BinTec router. Important information like system status of physical interfaces (e. g. ISDN line) and virtual interfaces (e. g. WAN partners) are easily available with ONE tool. A clear and complete overview of the load of a BinTec router's interfaces is possible at any time. The following illustration shows the state of a CM-PRI interface.



How does it work?

A status daemon collects information about the BinTec router and transmits it in the form of UDP packets to the LAN's broadcast address (default) or to a specified IP address. One packet per BinTec router interface and time interval, which is individually adjustable from 1 to 60 seconds, is sent. All physical interfaces and up to 100 virtual interfaces can be monitored unless the packet size of approx. 4000 bytes is not exceeded. A Windows application on your PC (available with BRICKware from Release 5.1.1) receives the packets and displays the information in different ways.

To activate the Activity Monitor you have to:

- Configure the BinTec router(s) to be monitored (this step is described here).
- Start the Windows application on your PC (the functionality of which is described in the document *BRICKware for Windows*).

Configuring the BinTec router

You can configure the BinTec router over the MIB table *ExtAdmin* or by using Setup Tool as follows.

The configuration is made in the menu **System** -> **External Activity Monitor**

BinTec router Setup Tool (SYSTEM)(ACTIVMON):External Activity Monitor	BinTec Communications AG MyRouter
Client IP Address	192.168.1.1
Client UDP Port	2107
Type	physical_virt
Update Interval (sec)	5
SAVE	CANCEL
Use <Space> to select	

Field	Meaning
Client UDP port (ExtAdminPort)	Number of port for the Activity Monitor (default: 2107, registered by the IANA - Internet Assigned Numbers Authority).

Field	Meaning
Client IP Address (<i>ExtAdminMonAddress</i>)	IP address to which the BinTec router sends the UDP packets. With the default value 255.255.255.255 the broadcast address of the first LAN interface is used. Be aware that if you enter the IP address of a WAN partner, connections liable to charges will be made at very regular intervals (default is every 5 seconds).
Type (<i>ExtAdminMonType</i>)	Type of information sent with the UDP packets to the Windows application. Possible values: <ul style="list-style-type: none"> • off: deactivates the Activity Monitor (default value) • physical: only information about physical interfaces • physical_virt: information about physical and virtual interfaces
Update Interval (sec) (<i>ExtAdminMonUpdate</i>)	Update time in seconds. Possible values: 0 to 60 (default: 5).

1.8 Web Based Monitoring

The BinTec router's operational state can be quickly polled via an HTTP server that has been implemented on the BinTec router. This server provides a status page which can be accessed from any WWW browser that supports HTML tables and the HTML 2.0 standard (i.e., Netscape's Mozilla or Microsoft Internet Explorer). The status page displays general system information, which licenses are installed, and current activity for each LAN or WAN interface.

Simply point a WWW browser at the BinTec router using the following URL. The http port is only required if it was changed from its default value of 80 .

http://<System Name>:<:HTTP Port Number>

The screenshot shows a Netscape browser window titled "mybrick: System Information - Netscape". The address bar contains "http://mybrick.mybrick.com". The page content includes the BinTec logo and the following sections:

System description

Type of System	BIANCA/BRICK-XS
System Name	mybrick
Location	Germany
Contact	sysadmin (sysadmin@mybrick.com)
Software	V.4.6 Rev. 1 from 97/09/30 00:00:00
System state	up and running for 6d 2h 34min

Software options

ip	ospf	stac	capi	bridge	x25	frame_relay	ipx
o.k.	nolicense	o.k.	o.k.	o.k.	o.k.	nolicense	o.k.

Hardware Interfaces

LAN	Ethernet	o.k.	
WAN	ISDN S0	o.k.	used 1, available 1
LOCAL			

You can [update](#) this page, see a list of [system tables](#), or [login](#) to the router.

For more information about BinTec products see <http://www.bintec.de>

SNMP-Table Browsing

The contents of the BinTec router's SNMP tables can be browsed via HTTP browsers using the "system tables" link from the main Status-Page. Initially this link displays a list

of all system tables found on the BinTec router. From there, individual system tables can be selected; the BinTec router creates the appropriate HTML pages on-the-fly showing the current contents of the respective variables.

The CGI ([Common Gateway Interface](#)) programs [htmlshow](#) and [snmpquery](#), are also available on the BinTec router and can be used to selectively display the values of one or more SNMP table objects.

CGI Program: htmlshow

The contents of one or more BinTec router SNMP variables can be selectively displayed to any WWW browser using the htmlshow program.

Note:



Only the "http" user may access the htmlshow program. The BinTec router authenticates htmlshow queries once per browser session by prompting the requestor for the http user's password. This value is defined in the **biboAdmHttpPassword** field of **bintecsec**.

The basic syntax for using htmlshow is as follows. Possible options are described below.

*separates CGI program
name from parameters* ↓

http: //<SysName> /htmlshow?<option=val>&<option=val>

*separates
parameter strings* ↑

htmlshow Options:

oid=snmp_oid

This option is mandatory and specifies an SNMP object identifier (OID) to display. *snmp_oid* is not case-sensitive. An OID may be specified in one of the following ways:

1. A symbolic object identifier name, i.e.
`.iso.org.dod.internet.mgmt.mib-2.interfaces.ifEntry.ifTable`
2. An numerical object identifier, i.e.
`.1.3.6.1.2.1.2.2.1`
3. A unique MIB-2 or BinTec MIB table or variable name, i.e.
`iftable`

Object identifiers starting with a period (“.”) are taken to be absolute object identifiers; otherwise a relative object identifier is assumed. Relative object identifiers are searched for relative to MIB-2, i.e. `.iso.org.dod.internet.mgmt.mib-2` or `.1.3.6.1.2.1`.

refresh*time=interval*

If *interval* is specified the display is updated every *interval* seconds. Entering 0 in the resulting text field disables automatic refresh updates.

orientation*=mode*

Defines the orientation of the output. “portrait” (default) or “landscape” mode may be specified.

If more than one object identifier is specified, the resulting tables or columns are printed side-by-side. The following URL was used to display the selected system variables shown on the following page:

```
http://mybrick/htmlshow?oid=isdChIsdnIfIndex&
oid=isdChState&oid=isdChReceivedOctets&
oid=isdChTransmitOctets&oid=isdChReceivedEr-
rors&
refresh=10
```

TIP: References to HTML pages generated by the BinTec router’s `htmlshow` program can be “bookmarked” for future reference. This will spare you the time of having to type long `htmlshow` queries (with the exception of the `http` password, all `htmlshow` options are saved in the bookmark)

mybrick: isdnchisdnifindex/isdnchstate/isdnchreceivedoctets/isdnchtransmitoctets/isdnchreceivederror - Netscape

File Edit View Go Communicator Help

Go to: <http://mybrick/htmlshow?oid=isdnchisdnifindex&oid=isdnchstate&oid=isdnchreceivedoctets&oid=isdnchtransmitoctets&oid>

isdnchisdnifindex / isdnchstate / isdnchreceivedoctets / isdnchtransmitoctets / isdnchreceivederrors / isdnchtransmitoctets / isdnchreceivederrors

Refresh time Orientation

isdnchisdnifindex	isdnchstate	isdnchreceivedoctets	isdnchtransmitoctets	isdnchreceivederrors	isdnchtransmitoctets	isdnchreceivederrors
IsdnIfIndex	State	ReceivedOctets	TransmitOctets	ReceivedErrors	TransmitOctets	ReceivedErrors
0 3000	0 connected	0 495332	0 117682	0 0	0 117682	0 0
1 3000	1 not_connected	1 1302726656	1 1292695552	1 5	1 1292695552	1 5
2 3000	2 not_connected	2 0	2 0	2 0	2 0	2 0

Go to the list of [system tables](#) , or back to the [home page](#)

Document Done

CGI Program: snmpquery

The contents of one or more selective SNMP object can also be polled from the BinTec router using the snmpquery program. This program is similar to the htmlshow program but it does not format its output as

HTML tables. (The output can still be read in any browser window). `snmpquery` is primarily intended for developers writing applications needing to access the BinTec router's SNMP tables via the network.

The syntax for `snmpquery` is shown below. Exactly one `oid=<value>` parameter must be present within each HTML request. .

*separates CGI program
name from parameter* ↘

`http://<SysName>/snmpquery?oid=<value>`

Specifying Object Identifiers:

`oid=value`

An SNMP OID (object identifier) can be specified using an absolute name or a shortname (the same names available from the SNMP shell). Values beginning with a dot, ".", are assumed to be absolute names. Values not beginning with a dot are assumed to be relative to MIB-2.

Additionally, objects can be specified in numerical or symbolic format (alphabetical characters uppercase, lowercase, or mixed). For example, any of the following `oid=<value>` parameters shown below could be used to retrieve the contents of the `tcp` static table.

`oid=.iso.org.dod.internet.mgmt.mib-2.tcp`
(absolute name – symbolic format)

`oid=.1.3.6.1.2.1.6`
(absolute name – numeric format)

`oid=tcp`
(relative name – symbolic format)

`oid=6`
(relative name – numeric format)

snmpquery Output

The output of the snmpquery program consists of a header line followed by the contents of each requested SNMP object.

The header line consists of a numeric HTTP result code and a status message. The following result codes are currently defined.

Result Code	Status Message
200	OK
400	Bad Request
401	Unauthorized
404	Not Found
500	Internal Server Error

These codes are described in detail in RFC 1945 (*HTTP 1.0*).

SNMP variable information is then displayed. Each line consists of three columns:

1. The object identifier (absolute name – numeric format) enclosed in quotation marks.
2. The SNMP variable type.
3. The variable's current value. (DisplayString objects are also displayed in quotation marks).

A HTML request for the system table would be displayed as follows:

```
200 OK
".1.3.6.1.2.1.1.1.0"      DisplayString      "BIANCA/BRICK-XM"
".1.3.6.1.2.1.1.2.0"      ObjectIdentifier
".1.3.6.1.2.1.1.3.0"      TimeTicks         23924186
".1.3.6.1.2.1.1.4.0"      DisplayString     "J.D.Smith (smith@sample.com)"
".1.3.6.1.2.1.1.5.0"      DisplayString     "mybrick"
".1.3.6.1.2.1.1.6.0"      DisplayString     "John's desktop"
```

“.1.3.6.1.2.1.1.7.0” Integer 12

1.9 User Accounts

You can log into the BinTec router using one of three different user IDs.

Admin Read Write

Passwords

For each user a separate password should be defined in the *bintecsec* table. Password information should be controlled. Default passwords (those set when your BinTec router arrives) are shown below.

Object Name	USER ID	Password
<i>biboAdmAdminCommunity</i>	admin	bintec
<i>biboAdmReadCommunity</i>	read	public
<i>biboAdmWriteCommunity</i>	write	public

The password (value of the respective Community object in *bintecsec*) defines the SNMP community name associated with all SNMP commands performed from the SNMP shell session.

User Rights

Each of the *bintecsec* users have a different level of access to the BinTec router's configuration information. As the system administrator you will almost always need to login as the admin user. The write and read users can be used to allow different levels of access to your system.

USER	Permission			
	System Table Editing	External System Commands	bintecsec Access	Setup Tool Access
admin	Read-Write	Execute	Read-Write	Execute
write	Read-Write	—	—	—
read	Read only	—	—	—

1.10 Other Passwords

HTTP Password

In addition to the SNMP community user passwords the bintecsec table contains the HTTP password for access to the BinTec router's main Status page.

By default the *biboAdmHttpPassword* object is set to **bintec**.

Note



The default HTTP password should be changed since it allows unrestricted read-access to all SNMP system tables on the BinTec router via HTTP.

RADIUS Secret

The RADIUS secret used by the BinTec router when contacting a configured RADIUS server (*biboAdmRadiusServer*) is also contained in *bintecsec*.

By default the *biboAdmRadiusSecret* is left empty.

1.11 System Software Updates

The BinTec router's system software is stored in flash RAM meaning that it can be easily updated allowing you to take advantage of newly developed/enhanced features not available when you purchased your BinTec router.

BinTec router system software updates are available via HTTP and FTP and are provided free of charge. You can always find the most recent software image for your BinTec router via our WWW server at: <http://www.bintec.de> For sites limited to character based connections software images are also available via our FTP site at: <ftp.bintec.de>.

1.11.1 What's Needed

To update the BinTec router's system software you will need the following:

- A BinTec router system software image,
- A direct serial port connection between your BinTec router and a PC where the software image is stored, –OR–
- An accessible (via a LAN or WAN interface) TFTP Server where the software image can be retrieved from.

1.11.2 Performing a System Software Update

Software Update via TFTP

1. Retrieve the system software image you wish to install using one of the URLs (FTP/HTTP) mentioned [above](#).
2. Place the software image in the TFTP server's TFTP directory.

Normally¹, this is : **C:\BRICK** for PCs running the *DIME Tools TFTP Server* application or **/tftpboot** on UNIX workstations.

3. For UNIX TFTP servers ensure that the image is world-readable.
4. Log into your BinTec router and issue the following command using the IP Address of your TFTP server and the image's filename.

update *IP_Address filename*

5. Enter **y** (yes) when asked: **perform update (y or n) ?**
6. Enter **y** (yes) when asked: **Reboot now (y or n) ?**

Software Update via XMODEM

1. Retrieve the system software image you wish to install using one of the URLs (FTP/HTTP) mentioned [above](#).
2. Place the software image on the PC your BinTec router's serial port is connected to. Preferably *BRICKware for Windows* should also be installed on this system.
3. Start the **BRICK at COM** terminal program for the serial port the BinTec router is attached to.
4. Now power up the BinTec router (or reboot the system using the **cmd=reboot** command if it's already running).
5. At the BOOTmonitor prompt press the spacebar to activate the BOOTmonitor.
6. Select menu item (3) and simply answer the questions as prompted on the screen. You'll need to specify the location of the software image and begin the file transfer.

1. The shown values are the defaults for most UNIX or PC systems, check your local configuration files to verify this location.

7. Once transferred you're given the option to update flash or write the image to memory. Select **u** update and then **b** boot the system.

1.12 BOOT Options on the BinTec router

When the BinTec router boots up, it performs several self tests. When the tests are finished the BinTec router optionally broadcasts BootP Requests via the first LAN interface if the IP address (for this interface is not configured).

1.12.1 The BOOTmonitor

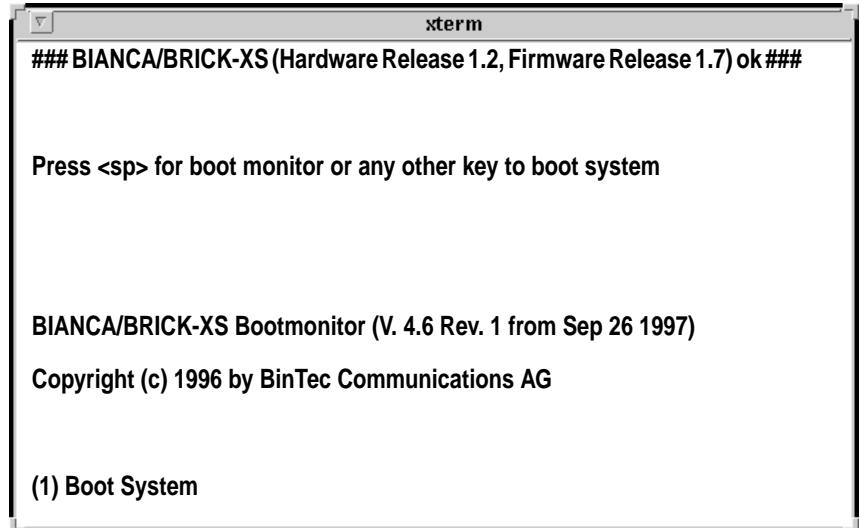
After the tests has been successfully completed, the BinTec router switches into BOOTmonitor mode and displays a prompt to the screen.

Note that the BOOTmonitor is only displayed on terminals connected. directly to the BinTec router's serial port. You will not see the BOTmonitor if connected via a LAN or WAN connection.

With the BOOTmonitor, you can easily perform firmware upgrades, test a new software release, or remove configuration files on your system.

To activate the BOOTmonitor the spacebar must be pressed within the first 4 seconds, otherwise the system continues with its normal boot procedure and switches into normal operation mode. Pressing the spacebar activates the BOOTmonitor as shown below. As long as the BOOTmoni-

tor is active (or awaiting keyboard input), all front panel LEDs will remain on.

A screenshot of a terminal window titled 'xterm'. The terminal displays the following text: '### BIANCA/BRICK-XS (Hardware Release 1.2, Firmware Release 1.7) ok###', 'Press <sp> for boot monitor or any other key to boot system', 'BIANCA/BRICK-XS Bootmonitor (V. 4.6 Rev. 1 from Sep 26 1997)', 'Copyright (c) 1996 by BinTec Communications AG', and '(1) Boot System'.

```
xterm
### BIANCA/BRICK-XS (Hardware Release 1.2, Firmware Release 1.7) ok###

Press <sp> for boot monitor or any other key to boot system

BIANCA/BRICK-XS Bootmonitor (V. 4.6 Rev. 1 from Sep 26 1997)
Copyright (c) 1996 by BinTec Communications AG

(1) Boot System
```

The commands from the BOOTmonitor menu are self guiding, informing/prompting you for confirmation along the way.

(1) Boot System

Select menu item (1) to load the compressed boot image from Flash into memory. This is the normal procedure performed at boot time.

(2) Software Update via TFTP

To upgrade the BinTec router's system software via a TFTP server select option (2). You will be prompted for the following pieces of information:

- IP Address of an accessible TFTP Server (where the image is stored).
- IP Address of the BinTec router
- The file name of the software image to retrieve.

Once you've entered the information and the image has been successfully retrieved you will be asked to confirm the update. Here, you have two options:

- (1.) Update Flash ROM
- (2.) Write image to RAM and boot it.

Note

Note that option (2) only loads the image into RAM and does not remove your existing boot image stored in Flash. With this option, you can test the new software release without removing your existing boot image. If the BinTec router is turned off, your old software release will be used upon a subsequent reboot.

(3) Software Update via XMODEM

You can upgrade system software via XMODEM over a serial connection with the BinTec router by selecting this option. You will be prompted to verify the baud rate to use over the serial connection. The time required to transfer the file will depend on the size of the file and baud rate you've chosen.

As when performing an update via TFTP you will then be prompted to confirm the update as follows:

- (1.) Update Flash ROM
- (2.) Write image to RAM and boot it.

(4) Delete Configuration

Select option (4) to return the BinTec router to its factory settings, as it arrived. All configuration files and BOOTmonitor parameters (see below) are removed.

(5) Default BOOTmonitor Parameters

Select option (5) from the menu to change the default settings used by the BOOTmonitor. These settings include:

- The baud rate used for serial connections.
- The LAN interface to use for TFTP file transfers.
- The Local IP address for the BinTec router.
- The IP address for the TFTP server.
- The system software image file to download.
- Automatic boot file retrieval over TFTP

The IP address settings defined here are used strictly for the BOOTmonitor and are not used for any IP routing functions on the BinTec router.

Note

If you change the baud rate, be sure that your terminal supports this rate, otherwise you may not be able to connect to the BinTec router. The default setting is set at 9600 baud, which is supported by practically all terminals.

Automatic booting over TFTP

The BinTec router can load its boot file via TFTP automatically at boot time by defining the appropriate settings in menu item (5). After setting the local and remote IP addresses, and the name of the system software image file to retrieve answer “yes” when asked the question:

```
Do you want to boot automatically from the TFTP
server (y or n):
```

1.12.2 Booting via BootP

The BinTec router's initial configuration information can be loaded remotely using a BootP server on the local network. This initial information normally includes the BinTec router's IP address and the name of its configuration file but may include other information. The server that provides this information may be a UNIX workstation running a bootpd process (see: [Setting Up a BootP Server](#)) or a PC running the included *DIME Tools BootP Server* program (see: [BRICKware for Windows](#)).

During every system startup, the BinTec router starts a BootP client process. Until an IP address is assigned, this process broadcasts standard BootP REQUEST packets every five seconds over the local network. Depending on how your BootP server is configured, the BinTec router can also load its configuration file remotely using TFTP. As soon as the IP address is received, the bootpd (client) process is ended.

Various information can be transmitted to the BinTec router using a BootP server. The BinTec routers BootP client process accepts the following BootP information (or TAGs) in accordance with the following Request For Comments (RFCs).

	TAG	RFC
Subnet Mask	1	1048
TimeServer	4	1048
TimeOffset	2	1048
IP Address	-	951
Host Name1	2	1048
Domain Name	15	1395
Domain Name Server	6	1048
Log Server	7	1048
TFTP Bootfile	-	951

Note

If the BootP server sends a hostname, domain name, or name server information, the BinTec router will accept this information (by setting the respective variables) only if this information hasn't already been set.

1.12.3 BootP Relay Agent

The BinTec router can also serve as a BootP Relay Agent for other hosts on the LAN. This is useful for stations that need to retrieve boot information remotely from a BootP server, but aren't on the same physical IP network as the server. If the BinTec router is on the same IP network as the station, it receives the stations BootP requests, and forwards them to server defined in *biboAdmBootpRelayServer*. See the section [BootP Relay Agent Settings](#) under [BOOTP and DHCP](#) in Chapter 7.

1.13 Other System Administration Tasks

1.13.1 Setting Up a BootP Server

To configure a BootP server on a UNIX workstation follow these instructions. The information shown below briefly describes setting up BootP to provide the BinTec router with basic IP settings (IP address, netmask, and name server's address). Refer to your local documentation for detailed description for your specific platform.

1. Edit (or create) the `/etc/bootptab` file to include the following lines:

```
brick: \  
  :ht=<the Hardware Type is usually "ether">:\   
  :ha=<the BinTec router's Hardware (or MAC) Address>:\   
  :ip=<the IP Address to use>:\   
  :sm=<the Subnet Mask to use>:\   
  :ds=<the Domain Name Server's IP Address>:
```

Note



The very first tag identifies the hostname this bootptab entry applies to. By default this is "brick" on systems where **sysName** hasn't been configured. If the system name is already configured specify that value here

2. You can start the bootpd process from the command line using:

On Solaris 2.5 and SunOS Systems:

```
/etc/bootpd -s
```

On Linux Systems:

```
/usr/sbin/bootpd -s
```

3. You may want to start the bootp daemon from the Internet Services daemon by adding the appropriate line to the `/etc/inetd.conf` file:

On Solaris 2.5 and SunOS Systems:

```
bootps dgram udp wait root /etc bootpd bootpd
```

On Linux Systems:

```
bootps dgram udp wait root /usr/ sbin/bootpd \  
bootpd bootptab
```

4. If you've added the bootps entry to /etc/inetd.conf as in step 3 you'll have to restart the inetd process for your changes to become effective.

On Solaris 2.5 Systems,

```
ps -ef |grep inetd  
kill -1 <pid>
```

On SunOS and Linux Systems,

```
ps -ax |grep inetd  
kill -1 <pid>
```

where <pid> is the process id of your running inetd process.

1.13.2 Setting up a TFTP Server

The [TFTP \(Trivial File Transfer Protocol\)](#) allows configuration files to be transferred to/from remote machines. The BinTec router implements TFTP allowing you to send and receive files to/from hosts where a TFTP server is running. The TFTP server may be a UNIX host or a PC running DIME Tools' TFTP Server application (see: [BRICKware for Windows](#)). A brief description of setting up a TFTP server on a UNIX workstation is covered below.

1. Allow the TFTP daemon to start. This is normally done by inserting one of the lines shown below in your /etc/inetd.conf file. Normally the correct entry is already present in the file and all you have to do

is uncomment it. Refer to your local documentation (inetd and tftpd) for more specific instructions.

On Solaris 2.5:

```
tftp dgram udp wait root /usr/sbin/in.tftpd \  
in.tftpd -s /tftpboot
```

On SunOS Systems:

```
tftp dgram udp wait root /usr/etc/in.tftpd \  
in.tftpd -s /tftpboot
```

On Linux Systems:

```
tftp dgram udp wait nobody /usr/sbin/tcpd \  
in.tftpd /tftpboot
```

2. Create the TFTP directory. You must separately create the TFTP directory (last field of the TFTP entry in *inetd.conf* shown above) and make it world readable using:

```
mkdir /tftpboot  
chmod 777 /tftpboot
```

3. Restart the inetd process. After you have added the above line to your local */etc/inetd.conf* file you must restart the inetd process. You must determine the process ID of inet daemon and restart the process. You can use the standard ps and kill commands as follows:

On Solaris 2.5 Systems

```
ps -ef |grep inetd  
kill -1 <pid>
```

On SunOS or Linux Systems:

```
ps -ax |grep inetd  
kill -1 <pid>
```

where <pid> is the process id of your running inetd process.

Remember that before you send TFTP files from the BinTec router to your (UNIX) TFTP server you must create the destination file in the TFTP directory and it must be world readable. This could be done using the commands:

```
touch /tftpboot/brick.cf
chmod 777 /tftpboot/brick.cf
```

Special Note:

TFTP Servers



Some UNIX TFTP server implementations (in particular **older BSD based systems**) do not reset the file length to 0 bytes prior to writing the TFTP file in response to a TFTP Write-Request; i.e., `cmd=put` or `cmd=state` is used.

This results in leftover data at the end of the TFTP file after the new data has been written. These files can not be processed by the BinTec router.

1.13.3 Setting Up a syslog Daemon

Log hosts configured on the BinTec router can be a PC running *DIME Tools Syslog Daemon* program or a UNIX workstation running a syslog daemon. This section briefly explains setting up an `/etc/syslog.conf` file for a UNIX workstation.

The exact format of this configuration file may be different on your UNIX platform, see your local documentation for more specific information.

1. As root edit the `/etc/syslog.conf` file to include the appropriate logging entry (see below). A typical logging entry that would save messages to a predefined file might look like this.

```
#facility.level      action
local0.info         /var/adm/brick.log
```

2. For actions that specify a log file, make sure you create the file and it has read-write permission for the syslog daemon.
3. Then as root stop and restart the syslog daemon.

On Solaris Systems

```
/etc/init.d/syslog start  
/etc/init.d/syslog stop
```

On SunOS Systems

```
kill -1 `cat /etc/syslog.pid`
```

On Linux Systems:

```
kill -1 `cat /var/run/syslogd.pid`
```

4. If you haven't already done so configure this host as a log host on the BinTec router. (See:).

Logging Entries in /etc/syslog.conf

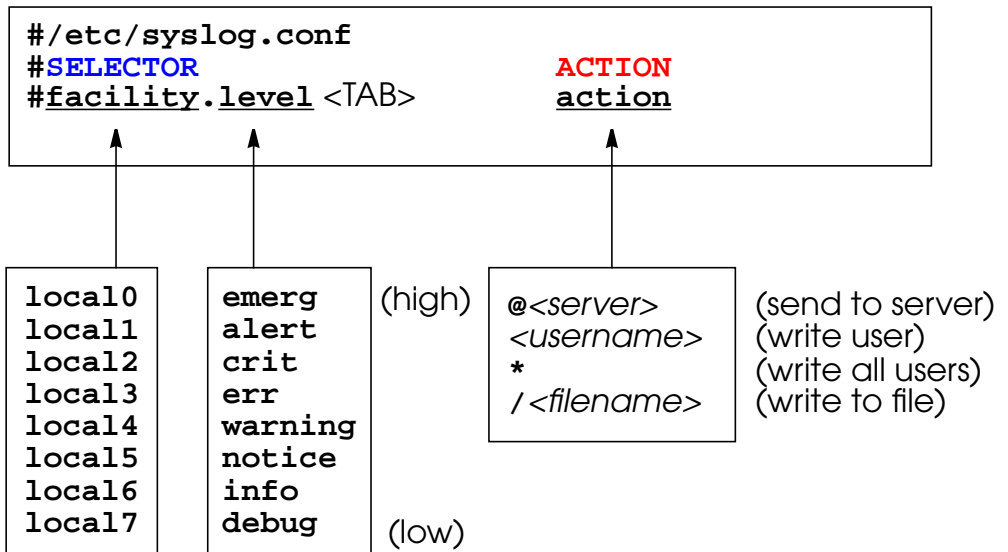
Logging entries in this file consists of two TAB-separated fields referred to as:

SELECTOR and **ACTION**

The **SELECTOR** field consists of a **facility.level** pair separated by a dot. (Actually, selector can contain multiple facility.level pairs separated by semi-colons, however, for the sake of simplicity we'll assume only one pair is being used.). The facility part identifies a system facility that a system message is received over; a sort of incoming port number if you will. The level part identifies the severity associated with the message.

The **ACTION** field identifies the action to take upon receiving a system message via this facility. Actions might include saving the system message to a file, writing to a specific user (if currently logged in), or forwarding the message to the syslogd of another host.

The facility and level of an incoming system message (i.e., sent from the BinTec router) must match both facility and level before the syslog daemon performs the action. The values that may be used in these field when configuring logging entries for the BinTec router are as follows:

**Note:**

On most systems the facility field must match the facility of the transmitting host or be "*".
 On most systems a level entry or X will match All messages (arriving on the respective facility) with levels \geq X. Some systems (Linux) support additional extensions in the level field to match level subsets.

1.13.4 Setting up a Time Server

The BinTec router acts as a Time Client and needs a Time Server to retrieve the time from. There are various possibilities: time can be retrieved from ISDN; the Time Server protocol via "Time Service UDP" is available on the Windows

software package, BRICKware; the time protocols “Time Service UDP/TCP are usually available on all Unix hosts; an XNTP Server package is freely available for PC/Unix servers, enabling the SNTP protocol via UDP.

Depending on the kind of server used, the BinTec router can retrieve the current time using any of the following four methods:

- Time Service (RFC 868) via UDP
- Time Service (RFC 868) via TCP
- Simple Network Time Protocol (SNTP) (RFC 1769)
Via individual Time Requests or Broadcasts: in the latter case, no explicit time requests are necessary, the Time Server automatically sends network broadcasts to all its time clients at regular intervals, thus saving packet traffic.
- ISDN D-channel (stack 0 only)

The following relevant SNMP variables are configured on the BinTec router in the *Admin* system table:

biboAdmTimeServer Specifies the IP-address of the Time Server in dot-format

biboAdmTimeOffset Specifies the time in seconds to add/subtract to the retrieved time. Values between -24 and +24 are assumed to be hours and are appropriately converted to seconds. Note that when time is retrieved from ISDN the offset must be set to zero.

biboAdmTimeProtocol Specifies the protocol to use to retrieve current time. Regarding the four methods noted above, the following protocols are possible.

- time_udp: Time Service (RFC 868) via UDP
- time_tcp: Time Service (RFC 868) via TCP

- `time_sntp`: SNTP (RFC 1769) via UDP
- `isdn`: ISDN D-Channel (stack 0 only)
- `none`: Disable time retrieval altogether

biboAdmTimeUpdate Specifies the interval in seconds at which current time should be updated/retrieved. As with Time Offset values between -24 and +24 are assumed to be hours and converted to seconds. For Protocol=`time_udp`, `time_tcp`, or `time_sntp` (if not in Broadcast mode) new requests are sent every *biboAdmTimeUpdate* seconds. When `isdn` is used, the current time is retrieved from the next ISDN connection established after *biboAdmTimeUpdate* seconds.

1.14 The Modem Function Module

1.14.1 V.90/K56flex Modem Function Module

This section describes the FML-8MOD Function Module.

1. We will start with a few introductory remarks concerning the V.90/K56flex technology in general and our modem module in particular.
2. Then we'll give an overview of the module hardware,
3. followed by a description of the modem configuration.
4. Finally, we'll give you a short example for setting up the modem module in an everyday situation, and explain how to trace a modem connection with the *BRICKware for Windows* software.

1.14.2 Introduction

V.90/K56flex Technology

The V.90/K56flex technology offers a new step up in modem speed. In conjunction with digital exchanges it is now possible to achieve data rates of up to 56kbps from central-site modems connected to the ISDN (e.g. internet service providers) to the client modem connected to the analogue telephone network (*downstream*). The other direction— from client to server (*upstream*)—still uses the V.34 standard with speeds of up to 33.6kbps.

This technology is especially useful for applications, where the data throughput is typically larger in the server→client direction (*downstream*), e.g. for internet providers.

FML-8MOD

BinTec's FML-8MOD—function module with eight modems—offers eight modems capable of all current modem standards up to and including V.90/K56flex. You can have up to four FML-8MOD modules installed in your BRICK-XL/XL2/XMP internally, thus offering up to 32 independent analog modems in connection with the FML-MODI modem connector module and a BIANCA/CM-PRI S_{2M} module.

Each modem on the FML-8MOD supports the following standards:

Standard	Description
V.90/K56flex	56,000, 54,000, 52,000, 50,000, 48,000, 46,000, 44,000, 42,000, 40,000, 38,000, 36,000, 34,000, or 32,000 bps downstream 33,600, 31,200, 28,800, 26,400, 24,000, 21,600, 19,200, 16,800, 14,400, 12,000, 9,600, 7,200, 4,800, or 2,400 bps upstream
V.34	33,600, 31,200, 28,800, 26,400, 24,000, 21,600, 19,200, 16,800, 14,400, 12,000, 9,600, 7,200, 4,800, or 2,400 bps
V.32bis	14,400, 12,000, 9,600, 7,200, or 4,800 bps
V.32	9,600, 7,200, or 4,800 bps
V.23	1,200 bps (1200/75, BTX)
V.22bis	2,400 or 1,200 bps
V.22	1,200 bps
Bell 212	1,200 bps
V.21	300 bps
Bell 103	300 bps
V.42 LAPM, MNP 2-4, 10	Error correction modes
V.42bis, MNP 5	Data compression

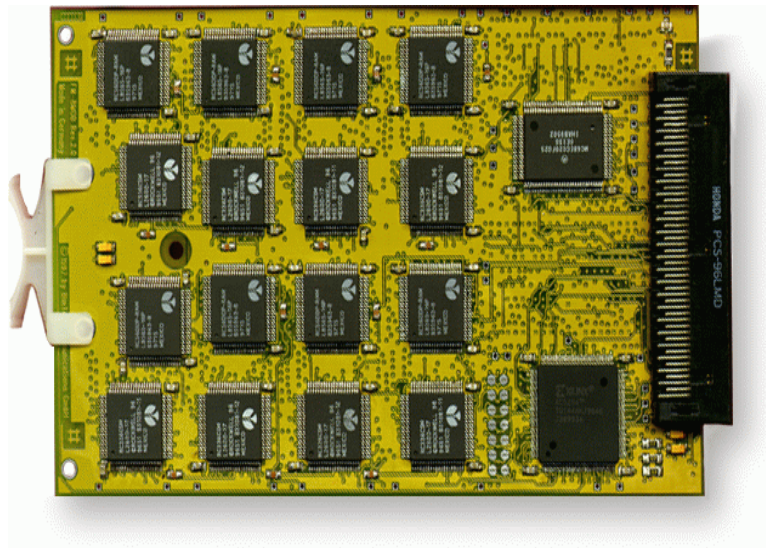
The modems are not bound to a certain B-channel, but are allocated to the next free channel as needed. This *dynamic resource allocation and distribution* technology (DRAD) provides for maximum flexibility.

You can easily update the system software for your modem modules by using the *modem* command (see p. 11). This al-

allows you to take advantage of new modem standards—without having to make any hardware modifications.

1.14.3 Hardware

The modem hardware consists of three different components. The FML-MODI (internal modem connector kit) comprises an SBus module which fits into slots 5, 6, or 7 (the extension slot) of your BinTec router, and a shuttle frame, which is installed in the lower part of your BinTec router and which holds up to four FML-8MOD modem modules..



Note The modem modules will always be installed by BinTec or by an authorized BinTec partner.

1.14.4 Software

Configuration

The modems are configurable over the *mdmProfileTable* in the MIB and the *isdnDispatchTable* and *biboPPPTable*. The following is a description of how to configure using the Setup Tool.

Main Page

On the main page of the Setup Tool there is a slot entry—Slot7—for the extension slot.

BIANCA/BRICK-XL SetupTool		BinTec Communications AG mybrick
Licenses	System	
Slot1:CM-BNC/TP, Ethernet	Slot4: CM-2BRI, ISDN S0, Unit 0 CM-2BRI, ISDN S0, Unit 1	
Slot2:CM-PRI, ISDN S2M	Slot5:	
Slot3:	Slot6:	
WAN Partner IP X.25 MODEM	Slot7: FM-MOD-56K/32	
Configuration Management Monitoring and Debugging		
Exit		
Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter		

In our example above slot 7 contains a modem connection module (FM-MOD-56K) with 32 modems available (four FML-8MOD modules are installed).

There is also the [MODEM] menu, see next section.

Modem Profiles

In the [MODEM] menu you can configure eight different modem profiles. All settings made in this menu show up in the *mdmProfileTable*.

In theory you could use only one profile, where all values are set to maximum—or auto, where applicable—and let the calling modem negotiate the values it needs.

This will work in most cases—only a few very old modems will not be able to negotiate the necessary values—but it takes much more time than connecting with the proper values in the first place.

Therefore you can use the profiles to grant different user groups different connection modes.

After starting the Setup Tool, go to the [MODEM] [Profile Configuration] menu, and select *Profile 1*. The default settings are shown in the figure below.

BIANCA/BRICK-XL Setup Tool [MODEM][PROFILE][EDIT]: Configure Profile		BinTec Communications AG mybrick
Name	Profile 1	
Description		
Modulation	V.34	
Error Correction	LAPM	
Automode	on	
Min Bps	300	
Max Receive Bps	33600	
Max Transmit Bps	33600	
V.42bis Compression	auto	
MNP5 Compression	auto	
SAVE	CANCEL	
Enter string, max length = 48 chars		

The fields have the following meanings:

Name Profile 1...8. Cannot be changed.

Note that Profile 1 is used as the default profile for modem connections, if no other profile is explicitly specified.



Description Descriptive string for this profile.
Modulation Modem standard to use, select with the space bar. Values range from K56flex down to Bell 103.
Error Correction Select the type of error correction to use.

Value	Meaning
none	Do not use any error correction.
required	First tries LAPM and then MNP5 error correction. If both fail, the modem will hang up.
auto	First tries LAPM and then MNP5 error correction. If both fail, the modem will not use error correction.
LAPM	Selects LAPM error correction. If this fails, the modem will hang up.
MNP5	Selects MNP5 error correction. If this fails, the modem will hang up.

Automode enable (on) or disable (off) negotiation of speed and modulation parameters.

MinBps The minimum baudrate you want to

use with this profile. You can set any speed supported by the current modulation (i.e. standard). Please refer to the table on page 65 for details. The connection will be released, if it cannot at least use the baudrate specified here.

MaxReceiveBps The maximum baudrate you want to use with this profile. You can set any speed supported by the current modulation (i.e. standard).

Max Transmit Bps Only needed in conjunction with the *K56flex* modulation. Sets the maximum transmit baudrate (»*downstream*«, server to client) you want to use with this profile.

V.42bisCompression enable (*auto*) or disable (*off*) negotiation for using V.42bis compression.

MNP5Compression enable (*auto*) or disable (*off*) negotiation for using MNP5 compression.

In addition to the above variables, the following variables can only be configured over the MIB table *mdmProfileTable*.

XmitLevel This object specifies the transmit attenuation in dB.

CDWaitTime This object specifies the amount of time in milliseconds the modem will wait for the appearance of the carrier. If the carrier does not appear within

- CDRespTime* this time period, the connection is terminated. This object specifies the amount of time a carrier must be present before it is recognized as a carrier.
- CDDiscTime* This object specifies the amount of time the carrier has to drop before the modem will assume the carrier to be lost.
- Retrain* Retrain addresses the common problem of poor or frequently changing line quality. The following values can be set:

Value	Meaning
retrain	Enables line-quality monitoring and auto-retrain: the modem controls the line quality and requests a retrain if required (no data transfer can take place during retrain).
off	Disables line-quality monitoring and auto-retrain: the modem neither controls the line quality nor requests a retrain.
fallback	Enables line-quality monitoring and fall-back/fall forward: the modem controls the line quality and falls back when the line quality is insufficient (a rate renegotiation to a lower speed) and falls forward when the line quality is sufficient (a rate renegotiation to a higher speed within the current modulation speeds). This is also the default value.

Incoming Call Answering

The [*Incoming Call Answering*] menu for all ISDN interfaces contains a list of an arbitrary number of entries rather than a mask with few possible variations. The settings from this menu show up in the *isdnDispatchTable*.

Note

Please refer to section *Partner Management* in chapter 4 of your *User's Guide* for more information on handling lists.

The entries in this list are used to distribute incoming ISDN calls received on this interface to different service items. The BinTec router distinguishes incoming calls based on the »Called Party's Number« transmitted with each ISDN call. Select one of your S_{2M} interfaces, then [*Incoming Call Answering*], and [*ADD*] to create a new list entry.

BIANCA/BRICK-XL Setup Tool		BinTec Communications AG	
[SLOT 2 ISDN S2M][INCOMING][ADD]: Conf. Incoming Call Answ.		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
PPP 64k	64kbps PPP data connection
PPP 56k	56kbps PPP data connection
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)
ppp_x75	This makes possible asynchronous PPP over X75 with PPP partners dialing in, even if these partners are authenticated inband (non-CLID).
Pots	only useful for V!CAS teleworking routers
PPP Modem Profile 1 ... 8	selects Modem Profile 1 ... 8 as configured in the [MODEM] menu

Value	Meaning
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 (<i>right to left</i> , this is the default), or from the left (<i>left to right (DDI)</i> , only useful for the Direct Dial In (DDI) feature of point-to-point ISDN accesses (<i>Anlagenanschluss</i> in Germany)).

1.14.5 WAN Partner / Outgoing Calls

Now to the [WAN Partner] [Advanced Settings] menu, where you configure ISDN partners.

The *Layer 1 Protocol* entry, which also shows up in the *bi-boPPPTable*, 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 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

Modem Utility

Included with the BinTec router's system software is the *modem* command. You can use this command to update the system software of your FML-MODI modem connector module, or to display the current operating status of all modems.

Software Updates

There are two prerequisites for performing a software update for your modem connector module:

1. You must have configured a TFTP host for your BinTec router (for instructions on how to do so

please refer to section *System Administration* of your User's Guide).

2. The new modem software image (available from our WWW server) must be located in the TFTP directory of your TFTP host.

Login to your BinTec router as user *admin* and then from the SNMP shell prompt issue the command:

```
modem update <TFTP host> <file name>
```

If you supplied the correct TFTP host and file name, you will see some screen output concerning the loading and verifying of the image file.

The update application will automatically detect all your modem connector modules and offer you to update each one individually.

```
Perform update for BIANCA/FM-MODI-56K in slot 7
(y or n)?
```

If you reply with »y« the update will be performed. This will take approximately 60 seconds.

After the update is complete you should reboot your BinTec router if you immediately want to use the new modem software.

Modem Status

To display the status of all modems issue the following command from the SNMP shell prompt of your BinTec router:

```
modem status
```

This will get you a display similar to the one below.

No	State	OBytes	Bytes	LastMessage
00	IDLE	280	2704	CONNECT 115200/K56/LAPM/NONE/38000:TX/31200:RX
01	IDLE	278	2701	CONNECT 115200/V34/LAPM/V42BIS/33600:TX/33600:RX

```

02 IDLE          18481      22233      CONNECT 115200/K56/LAPM/NONE/40000:TX/31200:RX
03 CALLING      0              0
04 CONNECTED    59635      64330      CONNECT 115200/V34/LAPM/NONE/33600:TX/33600:RX
05 CONNECTED    407         79         CONNECT 115200/K56/LAPM/V42BIS/36000:TX/31200:RX
06 CALLED       0              0
07 IDLE         0              0

```

The following table explains the possible modem states.

State	Description
IDLE	no modem activity
CALLING	outgoing call being set up
CALLED	incoming call being processed
CONNECTED	connection established,

1.14.6 Example Configuration

Central Site Modem Server

In this example we will show you how to set up your BinTec router as a modem server for *incoming* connections, where the callers receive their IP addresses and name servers from the BinTec router.

We assume that you are familiar with the basic operation of your BinTec router and the Setup Tool. For an introduction to these topics please refer to the *Getting Started* or *Los Geht's* manuals.

1. Login to your BinTec router and start the Setup Tool.

ISDN Partners

We'll start by adding a new ISDN partner for modem connections.

2. Go to the [WAN Partner] menu and select [ADD] to create a new partner entry.

BIANCA/BRICK-XL Setup Tool		BinTec Communications AG	
[WAN][ADD]: Configure WAN Partner		mybrick	
Partner Name	Mr. Smith		
Enabled Protocols	<X> IP < > IPX < > BRIDGE < > X.25		
Encapsulation	PPP		
Identify by Calling Number	no		
PPP Authentication Protocol	CHAP and PAP		
Partner PPP ID	smithbrick		
Local PPP ID	mybrick		
PPP Password	secret		
ISDN Ports to use	<X> Slot 3, ISDN S2M < > Slot 4, ISDN S0 (0)		
	< > Slot 4, ISDN S0 (1)		
ISDN Numbers >			
IP >			
Advanced Settings >			
	SAVE		CANCEL
Enter string, max length = 25 chars			

3. Give the partner a name and enter his PPP ID and the PPP Password to use with this partner.

Note



Make sure that *Identify by Calling Number* is set to **no**—analog modem calls usually do not contain a calling party number—and that you only use S_{2M} ports for modem connections.

You do not need to configure any ISDN Numbers at the moment.

4. Now go to the [*Advanced Settings >*] menu.

BIANCA/BRICK-XL Setup Tool		BinTec Communications AG	
[WAN][ADD][ADVANCED]: Advanced Partner Settings (Mr. Smith)		mybrick	
Callback	no		
Short Hold	3600		
Delay after Connection Failure	300		
Channel-Bundling	no		
RIP Send	none		
RIP Receive	none		
Van Jacobson Header Compression	on		
IP Accounting	off		
Dynamic IP-Address Server	on		
Layer 1 Protocol	ISDN 64 kbps		
Provider Configuration >			
	OK		CANCEL
Use <Space> to select			

The standard value for the *Short Hold* time (20 seconds) is too short for many modem connections—a typical modem call setup can easily last 30-50 seconds or longer—so you'll have to select a larger value. We chose **3600** seconds (1 hour), which—for all practical purposes—is equivalent to an infinite time, i.e. modem connections will only time out after 1 hour of inactivity. This is ok, because the connections to a central site server are usually initiated and closed by the client modem.

5. Set the *RIP Send* and *RIP Receive* fields to **none**.
6. Switch **on** *Van Jacobson Header Compression*, this will slightly improve your data throughput by reducing IP headers from 40 bytes to about 8 bytes per packet.
7. Switch **on** *Dynamic IP-Address Server*, this will allow the clients to get their IP addresses and name servers from your BinTec router. For information

on setting up a *Dynamic IP-Address Server* in the [IP] [Dynamic IP Addresses (Server Mode)], menu please refer to the User's Guide.

8. Confirm your settings with [OK] and [SAVE] this partner.

You could now go on and add a few more partners in the same manner.

Modem Profiles

Next we'll define new Modem Profiles for fast K56flex modem connections.

Note



As a default all eight modem profiles are set up for automatic speed and modulation negotiation, so that all modems from slow V.21 / 300bps types up to V.34 / 33,600bps types will be able to connect to your BinTec router.

1. Go to the [MODEM] [Profile Configuration] menu and select Profile 2. Leave Profile 1—which is the default profile for all modem connections where no

specific profile is specified—as it is for the time being.

BIANCA/BRICK-XL Setup Tool [MODEM][PROFILE][EDIT]: Configure Profile		BinTec Communications AG mybrick
Name	Profile 2	
Description	K56flex hi-speed	
Modulation	K56flex	
Error Correction	auto	
Automode	on	
Min Bps	28800	
Max Receive Bps	33600	
Max Transmit Bps	56000	
V.42bis Compression	auto	
MNP5 Compression	auto	
	SAVE	CANCEL
Enter string, max length = 48 chars		

2. Enter a description for this profile, select K56flex modulation, set the Error Correction to auto, and modify the data rates as indicated above. This profile will then only accept connections where at least 28,800bps are possible.

3. [SAVE] the profile.

You can also modify the other profiles to fit your demands.

Incoming Call Answering

1. Finally select the [CM-PRI, ISDN S2M] [Incoming Call Answering] [ADD] menu to configure a few tel-

ephone numbers for incoming modem connections.

BIANCA/BRICK-XL Setup Tool		BinTec Communications AG	
[SLOT 2 ISDN S2M][INCOMING][ADD]: Conf. Incoming Call Answ.		mybrick	
Item Number Mode	PPP Modem Profile 2 54302 left to right (DDI)		
SAVE		CANCEL	
Use <Space> to select			

2. In the Item field select the modem profile you want to use for this ISDN number, Modem Profile 2 (K56flex hi-speed) in our example.

Note:



Make sure that the number entered here exactly matches the called party number delivered with an incoming call. This is the number of your S_{2M} access plus the in-dialling number you want to use for this modem profile.

If in doubt, there is a rather easy way of finding out this number. *Do this only after you completed the rest of your configuration!* Leave the Setup Tool and issue the **debug all** command on your BinTec router. Then call the number of your S_{2M} access from any telephone, then dial a few more

digits and hang up. You will see an output similar to the following:

```
DEBUG/PPP: dialin from <> to local number <5430>
DEBUG/PPP: no matching dispatch table entry
DEBUG/PPP: dialin from <> to local number <54302>
DEBUG/PPP: ?: call accepted, call not identified by number
```

Press Ctrl-C to stop the debug output and note down the local number from the line immediately above the »no matching dispatch table entry« message. This is the way your number is signalled. In the [*Incoming Call Answering*] menu simply enter this number and append the in-dialling number to it.

3. For S_{2M} interfaces *Mode* must be set to **left to right (DDI)**.
4. [*SAVE*] the entry.
5. Now [*ADD*] another entry with a different *Number*, e.g. 54301, for Modem Profile 1.

Callers with K56flex modems can now use the number 54302, and all other callers can use the number 54301.

Enable Outgoing Calls

1. To enable outgoing modem connections to certain partners, e.g. for use with the Callback feature, go to the [*WAN Partner*] menu and select one of the partners already configured.
2. Go to the [*ISDN Numbers >*] menu and enter the number this partner can be reached at for modem calls.
3. You can then go to the [*Advanced Settings*] menu, and modify the settings as needed, e.g. enable Callback if desired, or reduce the Short Hold time, so that outgoing connections do not need one hour to time out, etc.

4. As a final step select the modem profile you want to use with this partner as *Layer 1 Protocol*.
5. [SAVE] your settings.

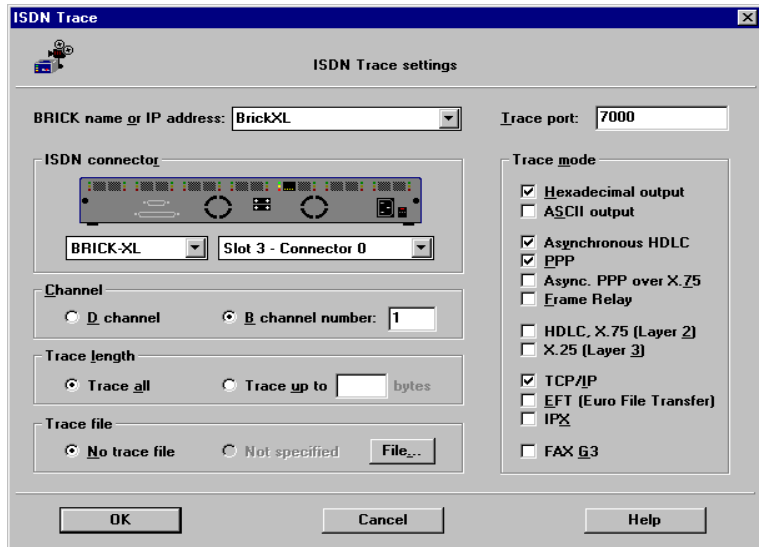
Now the partner can also be called using one of your BRICK-XL's modems.

1.14.7 Tracing a Modem Connection

You can use the *BRICKware* software package included on your Companion CD to trace modem connections.

TIP: Tracing modem connections can be especially useful when troubleshooting connection problems.

We'll assume that you have already installed *BRICKware* on your PC according to the on-line documentation. Then with your PC connected to the same LAN as your BinTec router, start *DIME Tools* and select *New ISDN Trace* from the *File* menu. This will get you the following dialog box.



Make sure to select *Asynchronous HDLC*, *PPP* and *TCP/IP* in the Trace mode area of the dialog.