http://www.bintec.de

Release Note BIANCA/BRICK-XS and BRICK-XS *Office*

March 23, 1998

New System Software: *Release 4.8 Revision 2*

This document describes the new features, enhancements, bug-fixes, and changes to the BIANCA / BRICK-XS System Software since Release 4.7 Revision 1.

New in 4.8.2	Upgrading System Software . . . . . 2 Bugfixes . . . . . 3 Important Note 4
What's New in Release 4.8.1	Features. . . . . 5 Encryption (MPPE) . . . . . 5 Virtual Private Networking and PPTP . . . . . 6 New Remote Multi CAPI Client (RMCC) . . . . . 6 MS-STAC Compression . . . . . 7 New Access List Methodology . . . . . 7 Local TCP/UDP Service Access Rules . . . . . 8 IP Multicasting Support for RIP V2. . . . . 8 Connected Line Identification Presentation. . . . 9 Login Accounting Messages . . . . . 10 IP Routing Algorithm Change. . . . . 10 New User Login Table. . . . . 12 RADIUS Enhancements . . . . . 13 Utility Enhancements . . . . . 14 Changes . . . . . 15 Bugfixes . . . . . 16 Detailed Feature Descriptions 19

Upgrading System Software

1. Retrieve the current system software image from BinTec's WWW server at http://www.bintec.de.

Please note that from release 4.8.1 on a new IP access list system is used. The old access lists will be automatically converted to the new system.
Before upgrading from a system running release 4.7.1 or older save your configuration to the flash ROM (using either *Save as boot configuration and exit* of Setup Tool, or **cmd=save**).
If you want to keep your old configuration you can also save it under a different name, e.g. **cmd=save path=config471**.
Then perform the upgrade as described here, check the converted access lists (see p. 7) and save the new configuration.

The access list conversion will ***not*** work if you load an older configuration file via TFTP. The BRICK will then contain ***no*** access list settings.

2. With this image you can upgrade the BIANCA/BRICK-XS with the **update** command from the SNMP shell via a remote host (i.e. using telnet, minipad, or isdnlogin) or by using the **BOOTmonitor** if you are logged in directly on the console. Information on using the BOOTmonitor can be found in the *BRICK-XS User's Guide* under *Firmware Upgrades*.
3. Once you've installed Release 4.8 Revision 2 you may want to retrieve the latest documentation (in Adobe's PDF format) which is also available from BinTec's FTP server at the address noted above.

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

What's New in Release 4.8

Release 4.8 Revision 2: Released: 23.03.98

Bugfixes

CAPI

- When using a CAPI application in transparent transfer mode the dataflow was occasionally interrupted, when the application sent too few data, resulting in a connection tear-down.

  This bug has been fixed.

IP Access Lists

- In rare cases the system rebooted when trying to convert complex access lists involving many interface-dependent conditions from a release ≤4.7.1 to release 4.8.1.

  This bug has been fixed.

PPP

- By mistake the *CallReference* field in the ***biboPPPLinkTable*** was not set in Release 4.8.1, therefore you could no longer monitor ISDN calls from the Setup Tool's [*Monitoring and Debugging*][*Interfaces*][*EXTENDED*] menu.

  This bug has been fixed.

- When setting up a DialUp-Networking PPP connection from a Windows PC to your BRICK the PC sometimes only acknowledged the connection set-up after about 30 seconds due to configuration inconsistencies—it seemed to be idle for this time.

  The set-up is now acknowledged much more quickly.

VPN / IP Tunnelling

- When setting up and tearing down a tunnelled IP connection several hundred times the BRICK occasionally rebooted due to an internal error.

  This bug has been fixed.

Important Note

IP Access Lists

- When configuring IP access filters depending on an IP address (e.g. *ipFilterSrcAddr* or *ipFilterDstAddr*) you must also specify an appropriate netmask.

  The IP address / netmask combination

  10.0.0.1/255.255.255.255

  means 10.0.0.1 only, while

  10.0.0.1/255.255.255.0

  means 10.0.0.1 through 10.0.0.255.

Release 4.8 Revision 1: Released: 09.03.98

Features: | Bugfixes: | Detailed Description:

Features

Encryption (MPPE)

The BRICK now supports user-data encryption according to the Microsoft Point-to-Point encryption protocol (MPPE). MPPE support allows encryption / decryption of user-data transmitted over PPP links. MPPE is negotiated at connect time as part of the CCP (Compression Control Protocol) sublayer of PPP and allows for additional security. MPPE is particularly useful for establishing secure Virtual Private Networking connections.

MPPE can be enabled for any PPP partner interface using the new ***biboPPPEncryption*** variable. Windows dial-up PPP partners may need to upgrade Dial-Up Networking software, see the note on page 6 regarding PPTP Support on Windows.

biboPPPEncryption

mppe_40	Encrypt data using a 40 bit session-key.
mppe_128	Encrypt data using a 128 bit session-key.
none	Disable encryption for this partner.

Securtiy

For security reasons the BRICK rejects PPP connections for partners where ***Encryption*** is enabled but couldn't be successfully negotiated at connect time. Also, once an encrypted link is established, (MPPE options were successfully negotiated) the BRICK will immediately terminate the link if the remote side attempts to disable encryption at any time during the connection.

NOTE: Since key generation is based upon the partners password data encryption is only possible if authentication (***biboPPPAuthentication*** = [**pap**|**chap**|**ms-chap**|**radius**]) is enabled. Also, if 128 bit encryption is desired the MS-CHAP authentication protocol must be used.

Virtual Private Networking and PPTP

With Release 4.8 the BRICK now supports Virtual Private Networking and the Point-to-Point Tunnelling Protocol (PPTP).

Virtual Private Networking is a recent development that allows you to both enhance connectivity and reduce communications costs while providing secure remote access to central site resources over the Internet. Using the BRICK as a VPN Server client-to-LAN or LAN-to-LAN PPTP connections (IP, IPX, or NetBEUI) can be “tunnelled” over the Internet. Allowing you to provide affordable yet secure remote access for distant or travelling workers, branch offices, or selected business partners.

By using the Internet as a transport medium both ends of the VPN avoid costly long distance charges and is only required to connect to their local Internet Service Provider.

NOTE: Virtual Private Networking support requires a separate license to be installed on the BRICK. Remote VPN clients will also require support for the PPTP protocol.

PPTP support for Windows 95 is a component of the Dial-Up Networking Upgrade 1.2. You can refer to Microsoft’s WWW site at http://www.microsoft.com/backoffice/communications/pptp.htm for details. PPTP is included on Windows NT 4.0 (Service Pack 3) and newer systems.

For a detailed description of VPNs and PPTP as well as the new SetupTool menus and MIB changes refer to the section Virtual Private Networking beginning on page 19.

New Remote Multi CAPI Client (RMCC)

BRICKware for Windows now supports CAPI connections over multiple BRICKs (Remote Multi CAPI Client, RMCC). You can now use the ISDN interfaces of all BRICKs available in your network for CAPI connections from one PC (applications allowing).

Note that RMCC is only available under Windows NT 4.0 (both server and workstation).

For a detailed description on installing and configuring RMCC please refer to section Remote Multi CAPI Client on page 40.

MS-STAC Compression

Starting in Release 4.8 the BRICK now also supports “Extended Mode” STAC LZS compression, known as Check Mode 4 in RFC 1974. Extended mode is the preferred mode on Windows 95 and NT systems and can now be successfully negotiated for dial-up connections on the BRICK.

MS STAC can be enabled for a PPP partner interface by setting the ***biboPPPCompression*** variable to **`ms_stac`**.

New Access List Methodology

Beginning in Release 4.8 the methodology used to configure IP Access Lists has changed. This new methodology is much more flexible and uses two new BRICK system tables: ***ipRuleTable*** and ***ipFilterTable***. IP Access Lists can still be configured via Setup Tool.

NOTE: With Release 4.8 installed your existing Access Lists (***ipAllowTable*** and ***ipDenyTable*** entries) are automatically converted to the new methodology and are saved to the two new system tables. If this configuration file is subsequently loaded using a software version older than 4.8 all IP Access List information will be lost. Therefore it is recommended to make a backup copy of your configuration files before upgrading. This can easily be done using the command **`cmd=save path=boot.47`** from the SNMP shell.

NOTE: Once your existing (pre-4.8) Access Lists are stored in the new ***ipRuleTable*** and ***ipFilterTable*** verify the new table entries are consistent with what your original access lists where designed to achieve. Once satisfied with the new entries make sure to save your configuration (using Setup Tool’s CONFIGURATION MANAGEMENT menu or the **`cmd=save`** command from the SNMP shell.)

For a detailed description of the new system tables and Setup Tool menus, see Access Lists beginning on page 46.

Local TCP/UDP Service Access Rules

For additional security, access to specific TCP or UDP services on the BRICK can now be controlled using the new ***localTcpAllowTable*** and ***localUdpAllowTable*** described below.

Access rules for BRICK TCP and UDP services are "Service" based. Access to a service can be based upon any combination of two criteria:

- The BRICK interface the TCP connection request (or UDP packet) arrived on.
- The IP address of the originating host.

The general rule for accepting / denying access to BRICK TCP / UDP services is as follows:

If an Access Rule exists for a TCP or UDP service then incoming connections to that service are allowed ONLY if:

1. The source address is 127.0.0.1, OR
2. No access rule exists for the requested service, OR
3. The incoming packet matches at least one Access Rule.
   i.e., source address = ***AllowAddr*** / ***AllowMask***, or
   source interface = ***AllowIfIndex***

For detailed information on using these new system tables see the section Local Service Access Rules beginning on page 45.

IP Multicasting Support for RIP V2

Support for IP Multicasting has been added in Release 4.8. In past releases multicast packets were not received by the BRICK.

Version 2 of the Routing Information Protocol (RIP) exchanges routing information with other hosts / routers by broadcasting or multicasting information over the network. With broadcasting each host on the network receives a copy of the packet. With multicasting RIP packets can be sent to selected groups of hosts using the class D address: 224.0.0.9.

The main advantage of multicasting is that network hosts not participating in RIP are spared the overhead of evaluating each received packet.

IP Multicasting support is configured in the ***ipExtIfTable*** using the ***RipSend*** and ***RipReceive*** fields for each interface.

ipExtRipSend

Defines the type of RIP messages that are sent over the interface. Default value: none

`ripV1`	Send version 1 RIP packets.
`ripV2`	Send version 2 RIP packets via broadcast.
`none`	Don't send RIP via this interface.
`ripV2mcast`	Send version 2 RIP packets via multicast.
`both`	A RIP version 1 packet is sent immediately followed by a RIP version 2 packet (both are broadcast).

NOTE: If ripv2mcast is configured, all routers participating in RIP must be multcasting capable.
(i.e., BRICKs running release 4.8 or newer)

ipExtRipReceive

Defines the type of RIP packets that may be received over the interface. Default value: none

`ripV1`	Accept RIPV1 packets only.
`ripV2`	Accept RIPV2 packets only. (broadcast and multicast are acceptable)
`both`	Accept RIPV1 and RIPV2 messages.
`none`	Don't accept RIP messages.

Connected Line Identification Presentation

The BRICK now supports the ISDN supplementary service Connected Line Identification Presentation, COLP.

This feature is important for CAPI 2.0 applications that utilize the *Connected Number* parameter of CONNECT_RESP or CONNECT_ACTIVE_IND messages.

Login Accounting Messages

Starting in Release 4.8 the BRICK now generates Login accounting messages that report information regarding login activity on the BRICK. An accounting message is generated each time a:

- Login session is started.
- Login session is closed.
- Login session failed due to incorrect password.

Accounting messages are shown in the table below where:
user = admin, read, or write
prog = CONSOLE, TELNET, ISDNLOGIN, or X.25PAD
time = the time the event occurred

biboAdmSyslogMessage	***~Level***
ACCT: LOGIN as <*user*> from <*prog*> at <*time*>	Info
ACCT: LOGOUT as <*user*> from <*prog*> at <*time*>	Info
ACCT: LOGIN FAILED as <*user*> from <*prog*> at <*time*>	Warning

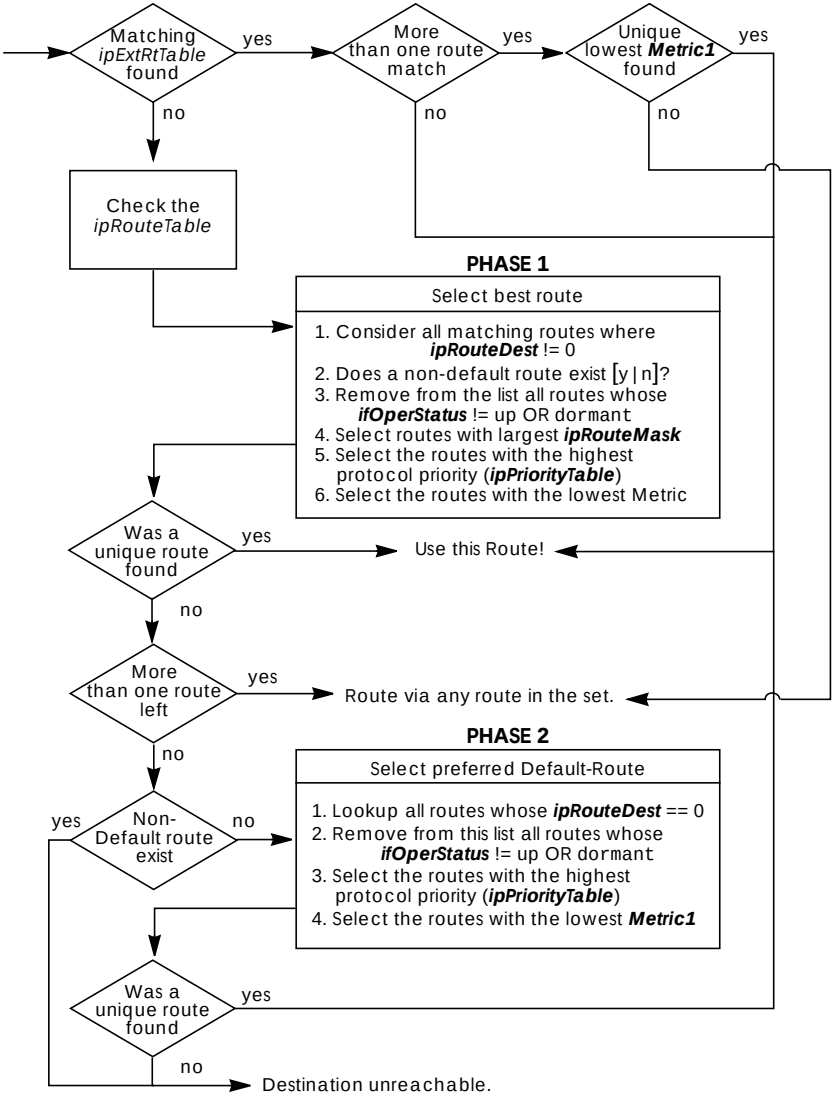
If the login session was from a program other than the console the accounting message also states the IP address of the source host, the X.25 address, or the calling party's number (if received via the ISDN).

IP Routing Algorithm Change

A change to the BRICK's internal IP routing algorithm was made that involves a shift in priority assignments. In past releases protocol priority considerations had a higher priority than the netmask. Although this change does affect route selection on the BRICK it won't affect most sites. The routing algorithm is described below.

Extended IP routes (the ***ipExtRtTable***) always take precedence over IP routes found in the ***ipRouteTable***. If an extended route isn't found the ***ipRouteTable*** is searched in two separate phases. Each phase consists of considering all IP routes, gradually reducing the list, and selecting the most desirable route.

Phase 2 is only entered if no routes were found during Phase 1. For the sake of clarity the new algorithm is outlined in detail in the flowchart shown below.



New User Login Table

The ***biboAdmLoginTable*** has been added to the system allowing you to create additional login accounts on the BRICK. For each login account a password and a special command (executed when the user logs in) can be assigned—similar to traditional User ID / Login-Shell control used on UNIX systems.

This is particularly useful for special-purpose connections such as having the login user automatically:

- open a telnet session to host a.b.c.d
- establish a UUCP connection to a specific host
- start a terminal connection to a specific host (i.e. Mailbox)

```
BRICK:> biboAdmLoginTable
inx  User(*rw)       Password(rw)     Command(rw)            State(-rw)

00   "checkmail"     "123"            "telnet 10.5.5.21"     valid
01   "uucp"                           "telnet -f 10.5.5.5 540" valid
```

Any external shell command (ping, telnet, minipad, isdnlogin, setup, etc.) can be used in the ***Command*** field. If the user enters Control-C, ***Command*** is stopped and the login connection is closed. Also, by using the "sh" command a BRICK SNMP shell connection can be started. For security reasons this table can only be viewed / changed by the admin user.

Limitations/Security Considerations

The following limitations and / or security aspects involving the use of the ***biboAdmLoginTable*** should be carefully considered:

- Data links that connect via special ***LoginTable*** entries generally may not be as efficient (throughput) as true "interface" connections since these connections are ultimately a sub-process of the initial login shell process which by default carries a lower priority than routing functions (refer to your documentation of the "p" command).
- Because the login-and-forward mechanism does require more system resources (memory in particular) than routed connections it isn't recommended for a great number of users.

- IP Access List Rules / Filters are not applied to sessions started via ***biboAdmLoginTable*** entries. Consider the "uucp" user account in the above example. Assuming the initial telnet session (from PC to BRICK) is allowed (no access restriction via ***localTcpAllowTable***) the telnet session to the host 10.5.5.5 will always be possible since this session is initiated by the BRICK locally.

RADIUS Enhancements

Two enhancements have been made to RADIUS in Release 4.8.

1. RADIUS accounting messages generated by the BRICK now contain NAS (Network Access Service) port information. On the BRICK the NAS port corresponds to the ISDN stack the connection was established on.
2. Two new BinTec-specific RADIUS extensions have been added in Release 4.8 which correspond to the variables ***Validate*** and ***DefaultPW*** (described below) that were added to the ***radiusServerTable***. A new dictionary file, Version 1.5, is also available from BinTec's WWW site.

   radiusSrvValidate

   This option is intended for bogus RADIUS servers, which send response messages with an incorrectly calculated MD5 checksum. All messages generated by the BRICK, however, will always use the proper authentication scheme. For security reasons, this option should always be left set to its default value of **`enabled`**.

`enabled`	Validate checksums from this server.
`disabled`	Do not verify checksums from this server.

   radiusSrvDefaultPW

   This is the default user password the BRICK sends when no password is available (for example, in requests for the calling number or boot requests). Some RADIUS servers rely on a configured USER- or CHAP-PASSWORD for any RADIUS request. The default value is an empty string.

Utility Enhancements

- **`telnet`**—The **`-f`** option has been added to telnet.

  The new syntax is as follows:

  `telnet` [**`-f`**] *host* [*port*]

  The **-f** option specifies that the telnet connection should be transparent. This option is especially useful for establishing connections to *non*-telnet ports such as uucp or smtp. See the example in the section New User Login Table.

- **`ifstat`**—The new **`-r`** option now displays the Access Rules that apply to the specified BRICK interface(s).

  The new syntax is as follows:

  `ifstat` [**`-lur`**] [<*interface*>]

  The **`-l`** option displays long output (normally only 12 characters of the ***ifDescr*** fields are shown) while the **`-u`** option displays status information for interfaces in the "up" state. For *interface* a numeric ***ifIndex*** or ***ifDescr*** may be used.

```
BRICK:> ifstat -r en1
01000 en1
Rul/Flt    Action          Descr          Conditions
001/001 deny M             telnet         daddr 192.168.12.1/32, dport 23
003/003 deny M             http           daddr 192.168.12.1/32, dport 80,
002/002 allow M            all-else
```

- **`netstat`**—The netstat command has two new options.

  The new syntax is as follows:

  `netstat` [**`-irp`** [<*interface*>] | **`-d`** <*dest_addr*>]

  With the <*interface*> parameter details about interfaces, routes, and partners can be limited to a selected interface. For *interface* a numeric ***ifIndex*** or ***ifDescr*** may be used. The **`-d`** option can be used to display IP routes to a destination address (specified in <*dest_addr*>).

NOTE: The **`-d`** option should not be confused with the **`rtlookup`** command. The **-d** option simply performs a

string match against all ***ipRouteTable*** entries and returns all routes whose ***ipRouteDest*** field starts with <*dest_addr*>.

```
BRICK:> netstat -i en1
Index Descr      Mtu    St  Ipkts  Ies    Opkts  Oes   Type Address
01000 en1        1500   up  15940  0      407    0     MAC 00:a0:f9:00:c0:11
                                                       IP  192.168.3.115
01001 en1-llc    1496   up  0      0      0      0     MAC  00:a0:f9:00:a0:11
01002 en1-snap   1492   up  0      0      0      0     MAC  00:a0:f9:00:a0:11
```

Changes

PPP

- The default PPP timeout for incoming connections (in-band authentication) was increased from 1000 to 3000 ms.
- ***biboPPPProfileTable***
  The default value for the ***AuthRadius*** variable in the ***biboPPPProfileTable*** has been changed from “inband” to “both”

Setup Tool

- In addition to Setup Tool’s new Virtual Private Networking menus the [ISDN Numbers] menu has been renamed to [WAN Numbers] in Release 4.8. Please be aware of this when references in your existing documentation are made to the old [ISDN Numbers] menu.

Bugfixes

CAPI

- A problem has been corrected which occurred when multiple BRICK capitraces were running simultaneously and characters were lost in the transfer.
- Information regarding the Channel Identification was incorrectly encoded in InfoInd messages sent by the BRICK.
- CAPI applications that took unusually long to confirm reception of data sometimes induced a problem on the BRICK that resulted in the system hanging.

IP

- In rare cases entries in the ***ipNatOutTable*** couldn't be deleted. This problem has been corrected.
- Incompatible ICMP implementation.
  If a routing table entry existed for the special REFUSE interface (***ifIndex***=0), "ICMP - destination unreachable" messages were sometimes incorrectly transmitted when establishing dialup connections. This has been fixed in Release 4.8 Revision 1.
- Problem with IP Fragmentation and NAT.
  On NAT interfaces IP packets that were fragmented (this only happens rarely since most protocols have a max packet size much smaller that the interface's MTU) were sometimes given an incorrect TTL value. This led to a problem where packets travelling over many hops were discarded and never reached their destinations.
- A problem involving NAT and improperly configured IP routing entries sometimes resulted in a system panic.

IPX

- A problem involving "piggy back updates" via RIP has been fixed.

OSPF

- IP routes added to the ***ipRouteTable*** that were based on OSPF External Advertisements received by the BRICK sometimes had an incorrect ***NextHop*** value.

PPP

- CCP Negotiation (STAC LZS Compression).
  A problem involving STAC negotiation options with CISCO IOS 11.2 routers hindered successful STAC LZS compression resulting in links without compression. This has been corrected.
- CCP (STAC LZS Compression)
  For interfaces where STAC compression was negotiated and the data to be transmitted couldn't be compressed the configured MTU size was sometimes exceeded and resulted in a disconnect during lengthy data transfers.
- CCP Negotiation on leased line interfaces.
  A problem involving CCP negotiation on leased bundle interfaces hindered the successful negotiation of STAC LZS compression for the link. PPP connections were established, but without compression. This has been fixed in Release 4.8.
- Leased Line interfaces.
  "PPP keep alive" packets (LCP Echo Request) are again transmitted at regular intervals regardless whether the interface is idle or active. Starting in release 4.6.2 keep alive transmissions where sent only after a 3 second idle time. This sometimes led to a state where the ***ifOperStatus*** never reached the down state once becoming idle.
- Data transfer errors sometimes occurred when using HDLC or LAPB encapsulation. The problem involved a rare state where packets were being received but the higher layer protocol instance hadn't acknowledged them before retransmission occurred. This has been corrected.

RADIUS

- RADIUS pings (used to verify connectivity) now work properly with the Steel-Belted RADIUS server.
- In previous releases connections that failed to setup via RADIUS were sometimes incorrectly logged (via an accounting message) as having been established.
- The BRICK sometimes hung when the data size contained in a RADIUS attribute was larger that the actual amount of data that was sent.
- RADIUS accounting messages are now RFC conformant in release 4.8.1.

Setup Tool

- In rare cases when HDLC or IP_LAPB encapsulation was selected in Setup Tool's [WAN][Partners] menu the [IP] submenu was no longer available.

X.25

- The BRICK sometimes transmitted Clear-packets in response to Call-packets received from X.25 LLC partners so that X.25 connections were not possible from these hosts.
- The Layer 2 settings (***L2WinSize***, ***L2RetryTime***, and ***L2RetryCounter***) configured for X.25 LLC partners in the ***x25linkPresetTable*** were not properly used when actually negotiating layer 2 parameters with LLC partners.

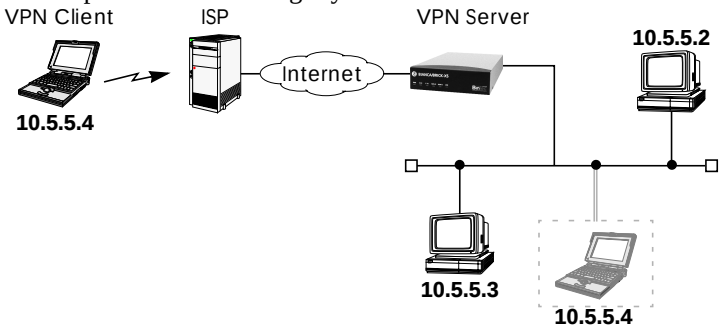
Detailed Feature Descriptions

Virtual Private Networking

Overview

A Virtual Private Network can be considered as a virtual Wide Area Network. It is *Virtual* in the sense that the network is not physical but is established on demand by software that establishes a link between a client and the server. VPNs are typically established over public (TCP / IP-based) data networks such as the Internet.

A VPN is also considered *Private* since user data transmitted over the link is typically encrypted. Windows 95 / NT based networks achieve this security via Microsoft's own Point-to-Point Encryption protocol, or MPPE. Since these VPN connections are encrypted (user data portion) network administrators can be assured that the use of the underlying public data network does not compromise data integrity.



The protocol that makes VPN possible is the Point-to-Point Tunnelling Protocol or PPTP. PPTP is an IETF standard described in RFC 1171.

The rest of this section describes the following in detail:

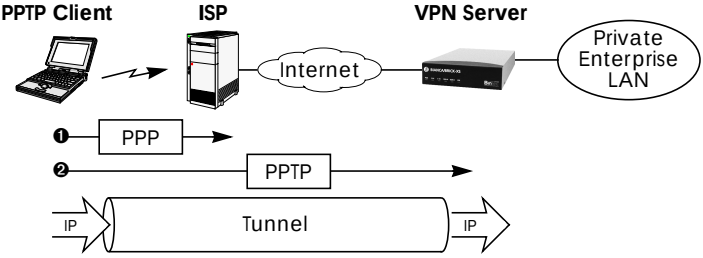
1. Tunnelling and PPTP
2. Authentication – Encryption – Compression
3. Example Client-to-LAN Configuration
4. Example LAN-to-LAN Configuration
5. SetupTool Menus for VPN / PPTP

Tunnelling and PPTP

Simplified, tunnelling is a method of encapsulating packets of one high layer protocol within the envelope of another high layer protocol (typically IP), “IP-over-IP” if you will. This technique also allows protocol data such as IPX and NetBEUI to be tunnelled via IP packets.

There are two commonly used scenarios for establishing VPN connections. The difference lies in which hosts involved in establishing the end-to-end connection support PPTP and which do not. Where PPTP support starts and stops also defines where the “tunnel” begins and ends.

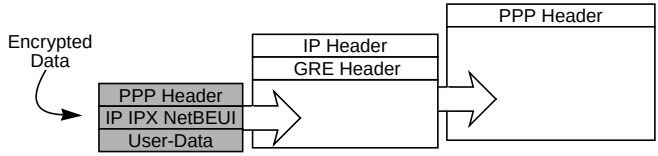
Scenario 1. PPTP Client–to–VPN Server



This is the most common scenario for PPTP. The remote client (mobile Win95 host) first establishes a standard PPP connection to a local ISP. The same client then initiates a second, logical connection, to the VPN Server. The ISP (and all intermediate Internet routers), unaware that it is participating in a VPN, simply routes IP packets from the PPTP Client.

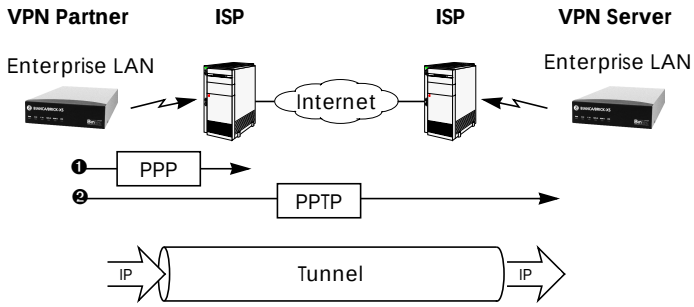
To hosts on the Private Enterprise LAN the remote PPTP Client appears as if it were directly connected to the LAN.

When sending data to the enterprise LAN the PPTP Client encapsulates PPP packets in the user-data field of the IP packet which is later unpacked by the VPN Server.



In the diagram above, GRE refers to the Generic Routing Encapsulation protocol. The GRE header identifies PPTP relevant functions and allows for efficient use of the link.

Scenario 2. LAN–to–LAN VPN



Here a Virtual Private Network that connects two enterprise LANs via the Internet is established via two VPN Servers. Either side may initiate a standard PPP link to a local ISP. Once the link is established the same server establishes a PPTP connection to the remote VPN server. Again, the ISP is unaware of its participation in the VPN.

All traffic routed via the ISP and destined for the remote LAN is encapsulated / unpacked by the respective VPN servers as mentioned in scenario 1.

Authentication – Encryption – Compression

In both scenarios above a second PPTP connection is established over an existing link. This second connection has its own PPP parameters (unique from those of the underlying link) with respect to user authentication, encryption, and compression.

Authentication

Both the ISP and the VPN Server will typically want to verify the initiating partner during connection establishment. Authentication is performed inband using PAP, CHAP, or MS-CHAP.

Data Encryption

Data encryption allows you to be sure that all user data transmitted over public data networks via a VPN is secure. The BRICK supports Microsoft's Point-to-Point Encryption protocol, or MPPE data encryption. Data encryption / decryption is performed at each end of the tunnel. Each host separately generates a *session-key* (40 or 128 bit key) using the respective partner's PPP password which is known to each host ahead of time.

NOTE: Since session-key generation is based upon the partner's password, data encryption is only possible if authentication (PAP, CHAP, or MS-CHAP) is enabled. Also, for 128 bit encryption the MS-CHAP authentication protocol is required (i.e., must be successfully negotiated at connect time.)

The Windows PPTP configuration dialoge includes an option for *password encryption*. This option applies to transmittal of the PPP password and does not apply to data encryption.

Compression

Data compression, depending on the data and the compression algorithm used, can increase performance over dial-up links as much as 30 fold (best case scenario using Stacker LZS). In both scenarios shown above, compression can be enabled for the initial PPP connection. Compression can also be enabled for PPTP links between BRICKs (Scenario 2: LAN–to–LAN VPN).

NOTE: The following limitation currently exists when combining compression + encryption for a PPTP link with Windows based hosts.

When the **Enable software compression** option is enabled in the **Server Types** tab (see Step 5.) Windows PPTP Clients offer EITHER MS-STAC Compression OR MPPE Encryption when tunnel parameters are negotiated. Currently, compression is only possible for the PPTP link if Encryption is set to "none" for the VPN partner interface on the BRICK (see the [VPN][ADD] menu on page 27).

Example Client-to-LAN Configuration

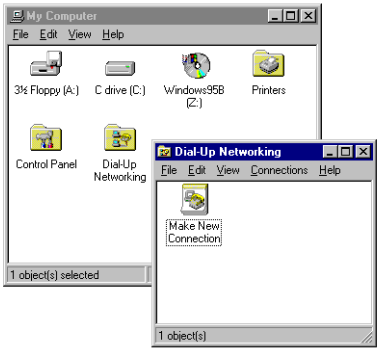
The Virtual Private Network shown in Scenario 1 on page 20 would be configured as follows.

Configure PPTP Client

Requirements: VPN Partners must support the PPTP protocol. For Windows 95 hosts this involves installing Winsock and Dial-Up Networking 1.2 Updates. Software updates and configuration information can be retrieved via Microsoft's web site at: http://www.microsoft.com/communications/pptpdownnow.htm

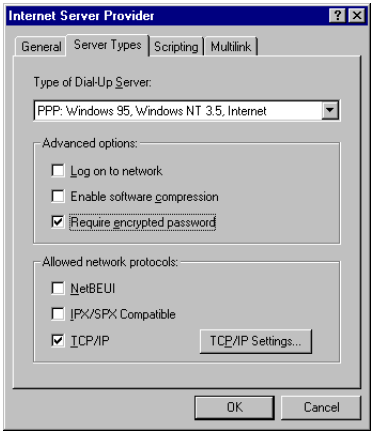
Configure PPP Link to the Internet Service Provider.

1. Open the Dial-Up Networking folder by double-clicking **My Computer**, and then **Dial-Up Networking** from the desktop.



2. Double-click the **Make New Connection** icon. In the resulting dialogue:
   – Specify a name for the ISP this host will be using.
   – Select a modem device to use for the ISP PPP link.
   – Then click the **Next** button.
3. Here you will need to enter the ISP's telephone number.
4. Click **Next>** and then **Finish**. A new icon will be added to the Dial-Up Networking folder. Right-click this icon and select **Properties** to display the properties window.

5. Click the **Server Types** tab.
   – In the **Type of Dial-Up Server:** field select:
   "PPP: Windows 95, Windows NT, Internet"
   – In the **Advanced options:** box
   Disable "Log on to network"
   Disable "Enable software compression"
   Enable "Require encrypted password"
   – In the **Allowed network protocols:** box
   Disable "NetBEUI"
   Disable "IPX"
   Enable "TCP / IP"



6. Click the **TCP/IP Settings...** button. Verify the IP address, name service, and compression settings are consistent with those required by the ISP and click **OK**.

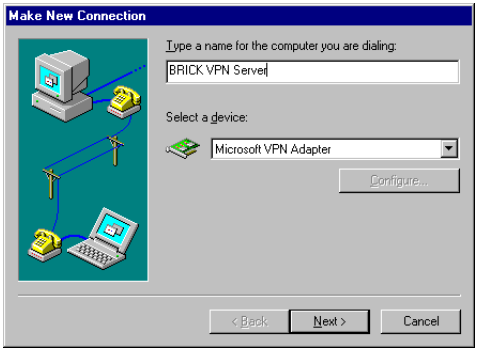
NOTE: In most cases the default settings in the **Scripting** and the **Multilink** tabs can be left untouched.

---

7. Click **OK** again. The initial PPP link to the Internet Service Provider is now configured. Proceed to the next section to configure the link to the BRICK VPN Server.

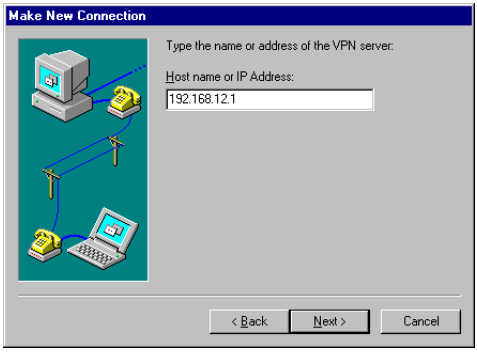
Configure the PPTP Link to the BRICK VPN Server.

1. From the **Dial-Up Networking** folder double-click the **Make New Connection** icon to configure the connection for the BRICK VPN Server.



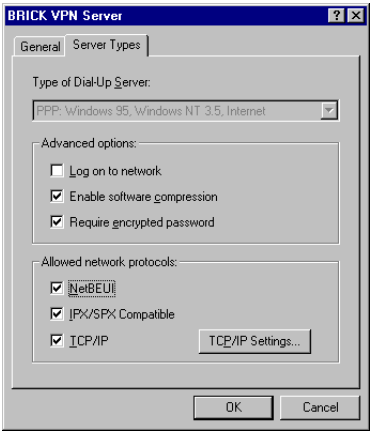
2. In the **Type a name for the computer you are dialing:** field specify a name for your BRICK VPN Server.
3. From the **Select a device:** drop menu select the device "Microsoft VPN Adapter" and click **Next>**.
   In the dialoge shown below enter the official IP address of the BRICK VPN Server.

NOTE: If the *Microsoft VPN Adapter* device is not available verify that version 1.2 (or newer) of Microsofts Dial-Up Networking software is installed.



4. Click **Next>** and the **Finish**. A new icon for the BRICK VPN Server will be added to the Dial-Up Networking folder.

5. In the Dial-Up Networking folder right-click the new BRICK VPN Server icon and select **Properties** to verify the connection settings.
6. Click the **Server Types** tab.
   – In the **Type of Dial-Up Server:** field select:
   "PPP: Windows 95, Windows NT, Internet"
   – In the **Advanced options**: box
   Enable "Log on to network" if hosts are required to register with the network.
   Enable "Enable software compression"
   Enable "Require encrypted password"
   – In the **Allowed network protocols**: box enable only those protocols this host will use to communicate with remote hosts on the central site LAN.
   At a minimum "TCP/IP" must be selected.



7. Click the **TCP/IP Settings...** button. Verify the IP address, name service, and compression settings are consistent with those on the BRICK and click **OK**. The settings used here must correspond to the respective BRICK VPN partner interface settings (see page 27).
8. Click **OK** again to accept the settings for the PPTP link. Once the respective BRICK partner interface is configured the Virtual Private Networking connection can be established as described on page 29.

Configure BRICK VPN Server

Requirements: A separate VPN license must be installed before the BRICK will support VPN connections. A VPN license can be purchased from BinTec Communications directly or from your local distributor.

Configure Link to the Internet Service Provider.

1. The link to the BRICK's ISP can be configured as a standard dial-up / leased PPP interface via Setup Tool's WAN Partners menu.

Configure the VPN Partner Interface

1. VPN partners are configured in the **VPN** menu. The settings below could be used for the VPN Partner (PPTP client) configured above.

```
BIANCA/BRICK-XS Setup Tool                BinTec Communications GmbH
[VPN][ADD]: Configure VPN Interface                          mybrick


Partner Name                  vpn1
Enabled Protocols             <X> IP  < > IPX < > BRIDGE
Encapsulation                 PPP
Encryption                    MPPE 40
Identify by Calling Address   no
PPP Authentication Protocol   MS-CHAP
Partner PPP ID                vpn1id
Local PPP ID                  mybrick
PPP Password                  vpn1pass


IP >
IPX >
Advanced Settings >

                      SAVE                    CANCEL

Enter string, max length = 25 chars
```

- In the **Encryption** field you may select MPPE encryption (40 bit or 128 bit session-key) or none.
- Disable ("no") the **Identify by Calling Address** option. This option can not be used since the BRICK will assign the PPTP client an IP address at connect time.

- In the **PPP Authentication Protocol** field select which authentication to use.

NOTE: If MPPE 128 encryption was selected the MS-CHAP protocol is required here.

- The **Partner PPP ID** and **PPP Password** fields define the values the VPN Partner must enter in the **User name** and **Password:** fields when establishing the VPN Connection.

2. Because Windows 95 PPTP clients expect the VPN server to assign them an IP address when the "tunnel" is established the **Dynamic IP Address Server** option in the ADVANCED SETTINGS sub menu must be enabled.

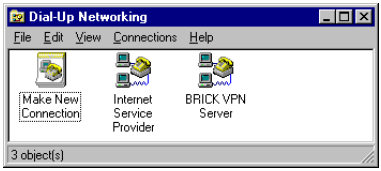
```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[VPN][ADD][ADVANCED]: Advanced Settings (vpn1)                 mybrick


Dynamic Name Server Negotiation  yes

RIP Send                         none
RIP Receive                      none

IP Accounting                    off
Dynamic IP-Address Server        on
Back Route Verify                off

                    OK                    CANCEL

Enter string, max length = 25 chars
```

For information on the other options available in this menu see the description of the [WAN PARTNERS][ADVANCED SETTINGS] menu your *User's Guide*.

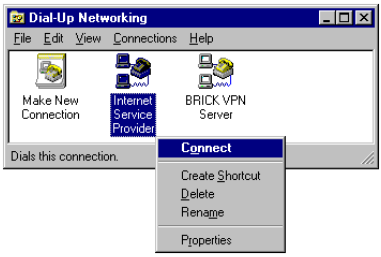
3. So that the BRICK can assign the PPTP client an IP address, make sure there are available IP addresses defined in the IP → Dynamic IP Addresses menu.

Connecting to the BRICK VPN Server

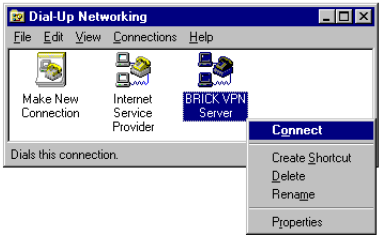
1. Open the Dial-Up Networking folder by double-clicking **My Computer**, and then **Dial-Up Networking**.



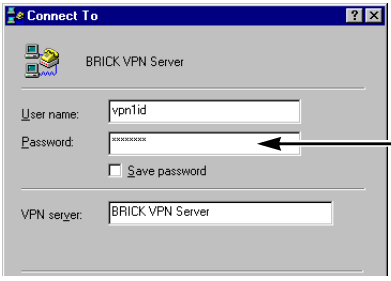
2. Right-click the Internet Server Provider icon, select **Connect** and enter the user / password assigned by the ISP.



3. After connecting to the ISP right-click the BRICK VPN Server icon and select **Connect**.

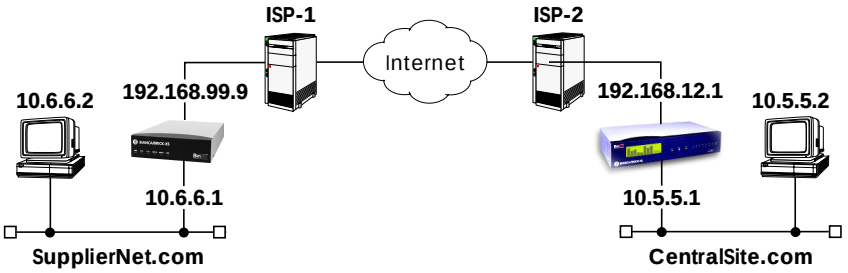


4. In the **Connect To** window shown below enter the PPP ID and PPP Password settings configured on the BRICK (see page 28) in the **User name** and **Password:** fields.



Example LAN-to-LAN Configuration

Two distant networks, a corporate central site LAN and a supplier or partner's network can be connected over the Internet via a Virtual Private Network using two BRICKs as follows.



Once both BRICKs are configured for Virtual Private Networking hosts on either LAN can connect to hosts on the remote LAN. All traffic that is routed between the two networks is encrypted (user-data encryption). Individual hosts are not required to support PPP or PPTP, the VPN remains transparent.

Configuration on SupplierNet BRICK

1. A separate license must be installed before Virtual Private Networking can be used. Verify the license is installed in Setup Tool's LICENSES menu.
   The status for "TUNNEL" must be "valid".
2. The link to the ISP-1 can be setup as a standard dial-up / leased PPP interface in the WAN PARTNER menu.
3. Configure the VPN Partner interface in the VPN menu. The VPN Partner interface for the BRICK-XL on CentralSite.com could be configured as follows.
   - Define a partner name (`csite`) and enable one or more protocols to support on the link.
   - In the **Encryption** field you may select MPPE encryption (40 bit or 128 bit session-key) or none. The options specified here must be the same for each partner.
   - Enable ("yes") the **Identify by Calling Address** option. The VPN partner will be identified by the IP address it uses when establishing the PPP link.

- In the **PPP Authentication Protocol** field select which authentication to use.

NOTE: If MPPE 128 encryption was selected the MS-CHAP protocol is required here.

- Set **Partner PPP ID** and **PPP Password** as needed.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[VPN][ADD]: Configure VPN Interface                            Supplier

Partner Name                  csite
Enabled Protocols             <X> IP  < > IPX < > BRIDGE
Encapsulation                 PPP
Encryption                    MPPE 40
Identify by Calling Address   yes
PPP Authentication Protocol   CHAP
Partner PPP ID                csiteid
Local PPP ID                  mybrick
PPP Password                  csitepass

IP >
IPX >
Advanced Settings >

                        SAVE                    CANCEL

Enter string, max length = 25 chars
```

4. In the **IP** menu you will need to define the IP addresses the VPN Partner will be using.
   - The **VPN Partner's IP Address** field for `csite` would be set to 192.168.12.1.
   - Under **via IP Interface** select the PPP interface for the local ISP. VPN connections to CentralSite.com may only be established over this interface.
   - Specify `csite`'s LAN address and netmask in the **Partner's LAN IP Address** / **Netmask** fields.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[VPN][ADD][IP]: IP Configurartion (csite)                      Supplier


VPN Partner's IP Address                  192.168.12.1
  via IP Interface                        ISP-1

Partner's LAN IP Address                  10.5.5.1
Partner's LAN Netmask                     255.0.0.0

                    SAVE                  CANCEL

Enter string, max length = 25 chars
```

5. In the ADVANCED SETTINGS sub menu the **Dynamic IP Address Server** option must be set to "off". Other options available there apply to the VPN interface and are described in chapter 4 of your *User's Guide* under the [WAN PARTNERS][ADVANCED SETTINGS] section.

Configuration on Central Site BRICK

1. A separate license must be installed before Virtual Private Networking can be used. Verify the license is installed in Setup Tool's LICENSES menu.
   The status for "TUNNEL" must be "valid".
2. The link to the ISP-2 can be setup as a standard dial-up / leased PPP interface in the WAN PARTNER menu.
3. Configure the VPN Partner interface in the VPN menu. The VPN Partner interface for the BRICK-XS on SupplierNet.com could be configured as follows.
   - Define a partner name (`SupplierNet`) and enable one or more protocols to support on the link.
   - In the **Encryption** field you may select MPPE encryption (40 bit or 128 bit session-key) or none. The options specified here must be the same for each partner.

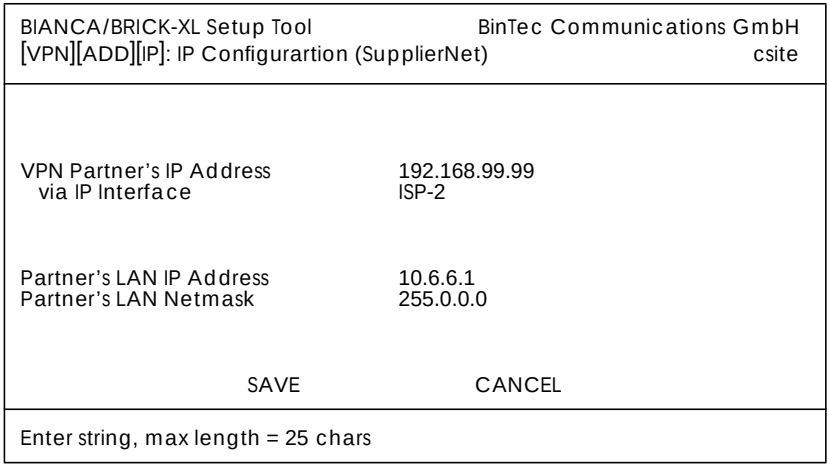
- Enable (“yes”) the **Identify by Calling Address** option. The VPN partner will be identified by the IP address it uses when establishing the PPP link.
- In the **PPP Authentication Protocol** field select which authentication to use.

NOTE: If MPPE 128 encryption was selected the MS-CHAP protocol is required here.

- Set **Partner PPP ID** and **PPP Password** as needed.

```
BIANCA/BRICK-XL Setup Tool                    BinTec Communications GmbH
[VPN][ADD]: Configure VPN Interface                                csite


Partner Name                  SupplierNet
Enabled Protocols             <X> IP  < > IPX < > BRIDGE
Encapsulation                 PPP
Encryption                    MPPE 40
Identify by Calling Address   yes
PPP Authentication Protocol   CHAP
Partner PPP ID                supplierid
Local PPP ID                  mybrick
PPP Password                  supplierpass

IP >
IPX >
Advanced Settings >

                        SAVE                  CANCEL

Enter string, max length = 25 chars
```

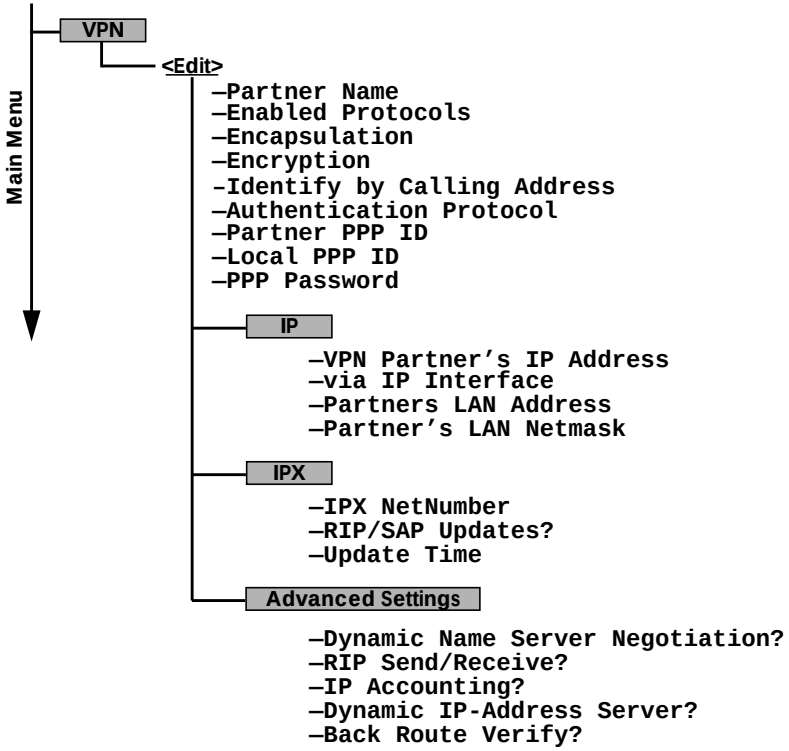
4. In the IP menu you will need to define the IP addresses the VPN Partner will be using.
   - The **VPN Partner’s IP Address** field for `SupplierNet` would be set to 192.168.99.99.
   - Under **via IP Interface** select the PPP interface for the local ISP. VPN connections to SupplierNet.com may only be established over this interface.
   - Specify `SupplierNet`’s LAN address and netmask in the **Partner’s LAN IP Address**/**Netmask** fields.



5. In the ADVANCED SETTINGS sub menu the **Dynamic IP Address Server** option must be set to "off". Other options available there apply to the VPN interface and are described in chapter 4 of your *User's Guide* under the [WAN PARTNERS][ADVANCED SETTINGS] section.

SetupTool Menus for VPN/PPTP

The VPN menu tree is displayed in the Setup Tools main menu when a valid VPN license is detected. The VPN menu is similar to the WAN Partners menu that you are already familiar with. Individual VPN submenus and fields are explained on the following pages.



VPN → ADD →

Use this menu to create Virtual Private Networking interfaces.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[VPN][ADD]: Configure VPN Interface                              mybrick


Partner Name                    tunnel
Enabled Protocols               <X> IP  < > IPX < > BRIDGE
Encapsulation                   PPP
Encryption                      none
Identify by Calling Address     no
PPP Authentication Protocol     CHAP + PAP + MS-CHAP
Partner PPP ID                  tunnel1-ppp-id
Local PPP ID                    brick
PPP Password                    tunnel1-ppp-pwd

IP >
IPX >
Advanced Settings >

                        SAVE                    CANCEL

Enter string, max length = 25 chars
```

Partner Name = The partner name assigned to this virtual interface.

Enabled Protocols = The protocols that may be routed over this interface.

Encapsulation = The type of encapsulation to use; currently PPP must be used.

Identify by Calling Address = This allows the BRICK to verify this VPN partner by its “calling IP Address”. This is the IP address the VPN partner can be reached at on the Internet (i.e., an official IP address).

PPP Authentication Protocol = The authentication protocol to use when authenticating this partner.

Partner PPP ID = The PPP ID that the VPN partner must identify itself with during PPP negotiation.

Local PPP ID = The BRICK’s PPP ID which is used during PPP negotiation with this VPN partner.

PPP Password = The password this VPN partner must use when challenged by the BRICK during PPP negotiation.

The VPN IP submenu defines IP address settings for the VPN partner interace.

Note: VPN partners will have two different IP addresses that define which network the host is on.

1. The Internet. This address must be an official address and defines where the host can be reached on the Internet. For the purposes of VPN, this address must be static (it may not be dynamically assigned by an ISP).
2. The VPN. The host's IP address on the local LAN.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[VPN][ADD][IP]: IP Configurartion (vpn1)                        mybrick

VPN Partner's IP Address                      192.168.12.99
  via IP Interface                            ISP

Partner's LAN IP Address                      192.168.13.99
Partner's LAN Netmask                         255.255.255.0

                        SAVE                  CANCEL

Enter string, max length = 25 chars
```

VPN Partner's IP Address = The VPN partner's IP address where it can be reached at on the Internet.

via IP Interface = The IP interface that packets received from this VPN partner will be received on. This will typically be the interface to the Internet Service Provider.

Partner's LAN IP Address = The VPN partner's LAN address.

Partner's LAN Netmask = The netmask the partner uses on it's LAN. If left blank, a standard netmask for the respective network class will be used.

VPN → ADD → IPX →

The VPN IPX submenu defines IPX relevant settings for VPN partner interfaces that support IPX.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[VPN][ADD][IP]: IP Configurartion (tunnel)                      mybrick

IPX NetNumber              0
Send RIP/SAP Updates       triggered + piggyback
Update Time                60

                      SAVE                CANCEL

Enter hex number range 0..fffffffe
```

IPX NetNumber = The IPX network number of the network link (the PPTP link). This is required by some IPX routers.

Send RIP/SAP Updates = Determines how often RIP and SAP packets are tranmitted to this VPN partner. The possible options are the same as those defined in the menu, see chapter 4 of the *User's Guide* for additional information.

Update Time = Determines how often (in seconds) periodic updates are sent to this VPN partner.

VPN → ADD → ADVANCED SETTINGS

The settings defined here are similar to the [WAN PARTNERS][ADVANCED SETTINGS] menu but apply specifically to an VPN partner interface.

```
BIANCA/BRICK-XS Setup Tool                BinTec Communications GmbH
[VPN][ADD][ADVANCED]: Advanced Settings (tunnel)              mybrick


Dynamic Name Server Negotiation  yes

RIP Send                         none
RIP Receive                      none

IP Accounting                    off
Dynamic IP-Address Server        off
Back Route Verify                off

                OK                      CANCEL

Enter string, max length = 25 chars
```

Dynamic Name Server Negotiation = Defines whether (and how) the name server's address is configured.

RIP Send/Receive = Defines the which version of RIP packets to exchange with this partner.

IP Accounting = Enable / disable generation of IP accounting messages for this partner. When enabled, an accounting message is generated (and written in ***biboAdmSyslogTable***) which contains detailed information regarding connection activity for this partner.

Dynamic IP-Address Server = Defines whether or not the BRICK should assign this partner an available IP address from the IP address pool.

Back Route Verify = When enabled the BRICK verifies that the return route for all packets received from this partner interface uses the same interface the packet arrived on.

Remote Multi CAPI Client

What is it?

The Remote Multi CAPI Client (RMCC) enables you to use multiple BRICKs for CAPI connections from one PC running Windows NT 4.0. The RMCC allows your CAPI applications to take advantage of all ISDN controllers available through one or more BRICKs on the LAN. By providing a pool of available ISDN controllers, access to the ISDN (whether via a remote or local controller) remains transparent to the application.

This can e.g. be useful for fax server applications which can then send and receive several faxes at the same time.

To make use of the RMCC feature your 32bit CAPI 2.0 application must be able to address several different CAPI controllers at the same time.

RMCC is also able to automatically reconnect to a BRICK after it rebooted, i.e. you do not manually have to stop and restart all CAPI applications if the BRICK reboots.

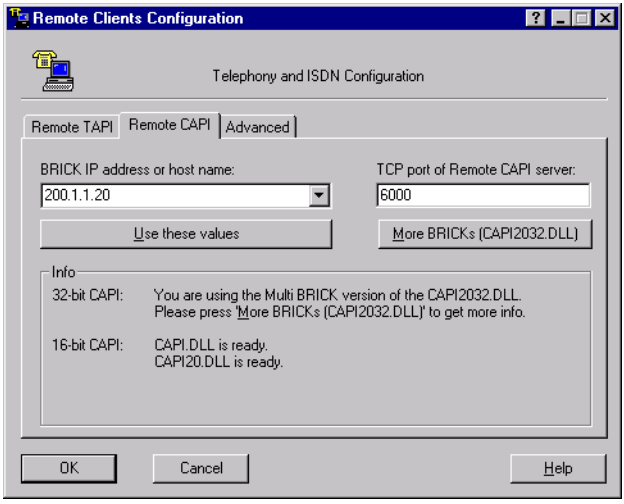
Installation

Simply install the latest version of BRICKware for Windows (4.8. Rev. 1). If you have Windows NT 4.0 running on your PC the Remote Multi CAPI Client (an enhanced version of the CAPI2032.DLL) will be installed automatically.

The 16bit versions of CAPI 1.1 (CAPI.DLL) and of CAPI 2.0 (CAPI20.DLL) are, of course, still available for use with one BRICK at a time.

Configuration

You can configure the Remote Multi CAPI Client from the TAPI and CAPI Configuration program which is located in the BRICKware program group.



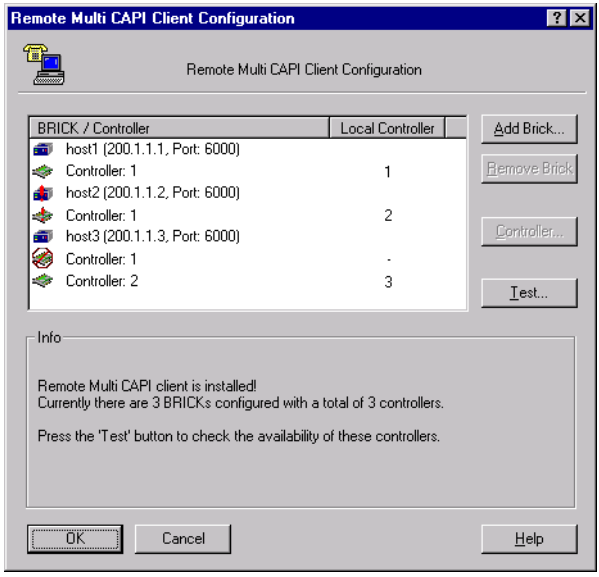
Make sure to close all CAPI applications before changing your CAPI configuration.

If you only want to use one BRICK for CAPI applications, configure it as usual by entering its hostname or IP address and its CAPI TCP port in the appropriate fields.

The BRICK configured in this dialog will be used for 16bit CAPI applications and is also used initially for 32bit CAPI applications.

If you want to use two or more BRICKs simultaneously, press the new *More BRICKs* button.

You will then get a list of all BRICKs currently configured and their controllers (for 32bit CAPI applications).



The list will initially be empty (unless you already configured a BRICK on the main page).

If you select a BRICK from this list, the Info field in the lower half of the dialog box will display the number of controllers available on this BRICK, whether a CAPI license is installed, the system software revision, and the serial number.

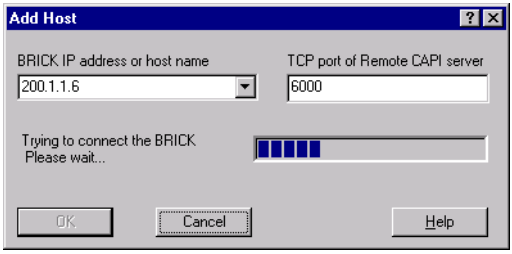
If you select a Controller from this list, the Info fieldwill display the number of B channels available from this controller, whether DTMF tones are supported, and the supported B1-layer protocols.

Changes made to this list (for 32bit CAPI applications) will *not* automatically affect the settings made for 16bit CAPI applications in the main dialog.

Add BRICK

To add a BRICK click the *Add BRICK* button. Enter its hostname or IP address and its CAPI TCP port in the appropriate fields.

When you click on the *OK* button to confirm your entries, the application will try to establish a connection to the BRICK and retrieve information on the number of controllers available on this BRICK and on its system software release and serial number.



This may take a couple of seconds. If the connection fails make sure the BRICK is switched on, connected to the network, the IP address and CAPI TCP port are correct, and try again.

All controllers of the BRICK will be added to the list of available controllers and will automatically be assigned a new local controller number.

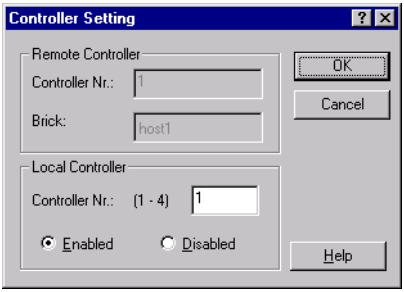
The list of local controller numbers always starts with controller #1 and does not contain any gaps, e.g., if you remove a BRICK or disable a controller the remaining controllers are automatically renumbered.

Remove BRICK

Removes the selected BRICK and its controllers from the list of available controllers.

Configure Controller

By double-clicking on a controller (or first clicking the controller and then clicking the *Controller…* button) you get the following dialog.



Here you can assign a different local controller number to the controller, or Enable or Disable it for CAPI connections.

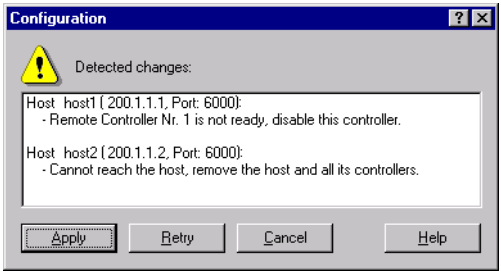
Test

After changing your configuration you should click the *Test* button. The program will try to verify the data of all BRICKs and controllers currently configured.

You can interrupt the test with the *Stop Test* button.

If the test reports OK you can save your configuration with the *OK* button.

If any errors are detected the program will display a dialog box similar to the following, suggesting what to do in the case of the detected discrepancies.



Click on the *Apply* button to make the suggested changes.

Local Service Access Rules

localTcpAllowTable

The ***localTcpAllowTable*** defines access rules for TCP services. Each entry defines an access rule for a specific TCP service. Controllable TCP services (***Service*** field) include:

telnet	trace	snmp	capi
tapi	rfc1086	http	

The ***localTcpAllowTable*** entries shown below:

1. Limit access to the BRICK's HTTP service to a single host (at IP address 192.168.12.2), and
2. Limits access to the BRICK's telnet service to all hosts on the 192.168.5.0 network.

```
BRICK: > localTcpAllowTable

inx  AddrMode(-rw)     Addr(*rw)        Mask(rw)          IfMode(rw)
     IfIndex(rw)       Service(rw)

00   verify            192.168.12.2     255.255.255.255   dont_verify
     0                 http
01   verify            192.168.5.0      255.255.255.0     dont_verify
     0                 telnet
```

localUdpAllowTable

The ***localUdpAllowTable*** defines access rules for UDP services. Each entry defines an access rule for a specific UDP service. Controllable BRICK UDP services (***Service)*** include:

snmp	rip	bootps	dns

The following ***localUdpAllowTable*** entry limits access to the SNMP service on the BRICK to hosts on the LAN.

```
BRICK:> localUdpAllowTable

inx  AddrMode(-rw)     Addr(*rw)        Mask(rw)          IfMode(rw)
     IfIndex(rw)       Service(rw)

00   dont_verify       0.0.0.0          0.0.0.0           verify
     1000              snmp
```

Access Lists

The new IP Access List methodology used on the BRICK is based upon a concept of Rules, Filters, and so-called Chains.

Suggested Method for configuring Acces Lists

Because the potential danger exists of "locking oneself out of the system" when configuring IP Access Lists the following order of events should be used.

1. Define the set of filters to use.
2. Disable access lists for all interfaces by setting the "FirstRule" to "0 (no access rules)".
3. Define the complete set of rules.
4. Enable the rule(s) for the desired interfaces.

Access List Methodolgoy

An Access Filter simply describes a subset of IP traffic and may be based upon one or more of the following attributes.

- Source and / or Destination IP address.
- Source and / or Destination Port.
- Source and / or Destination Protocol.

An Access Rule defines an:

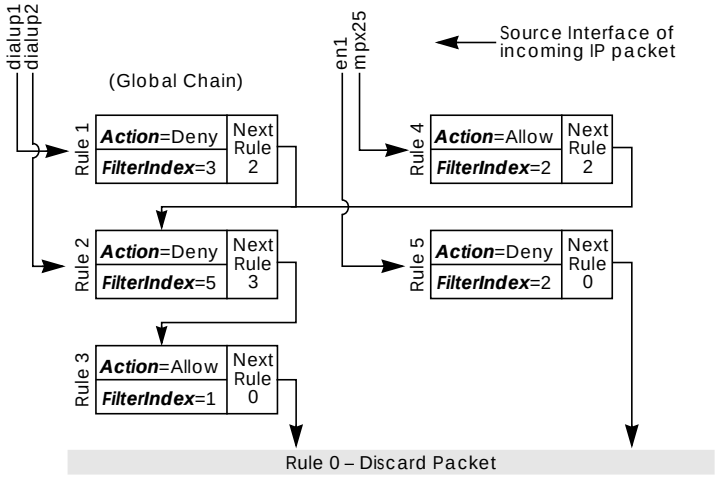
1. Access Filter to compare the packet to.
2. Action to take if a packet matches / doesn't-match a filter.
3. Index of the next rule to use if no action was taken.

Each Rule references a NextRule allowing different *Chains* (sequence of Rules) to be defined. For each interface a separate starting rule must be defined (via the ***ipExtIfRuleIndex*** field) that determines which Rule chain is applied. Rule 1 has special meaning; it is used by default for all newly created interfaces.

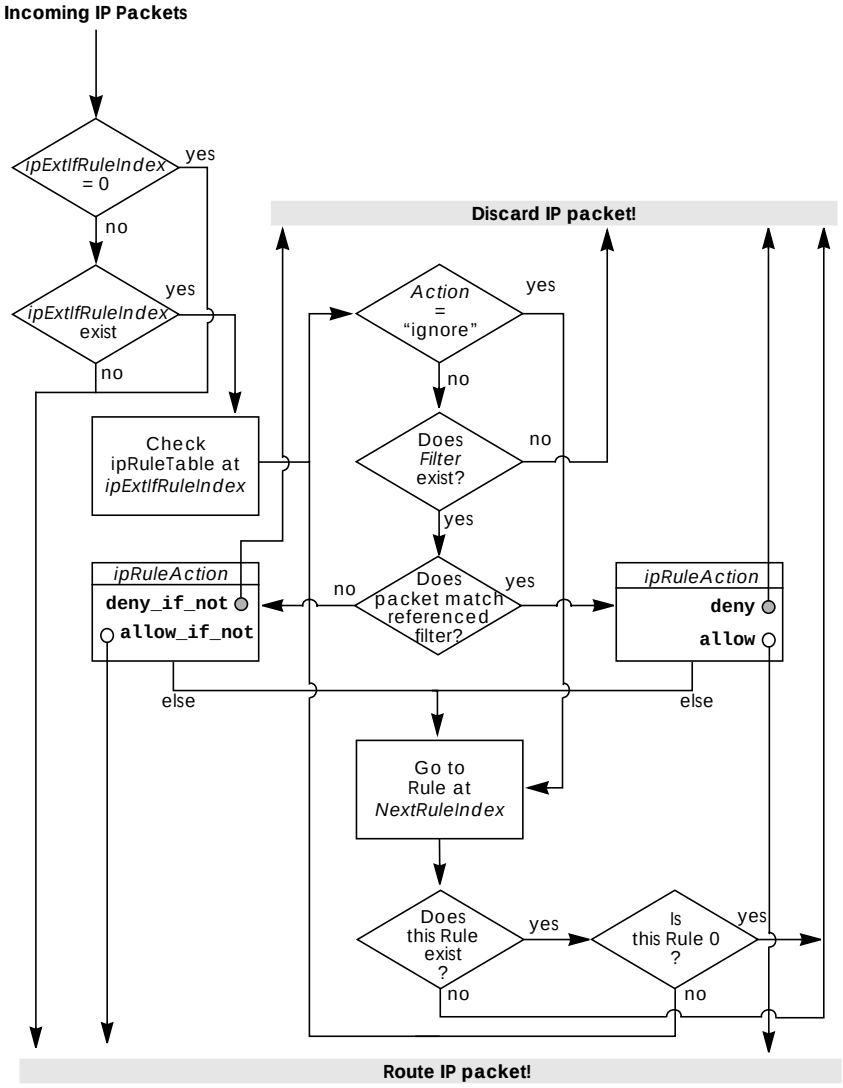
Rules are applied until one of the following events occur:

- The packet matches and the ***Action*** is "match" based OR the packet doesn't match and the ***Action*** is "if_not" based.
- The packet is discarded if the end of the chain or Rule 0 is reached.

In the diagram below, packets arriving via the “dialup1” interface are compared to Rules 1–2–3 while packets arriving on the “mpx25” are applied to Rules 4–2–3.



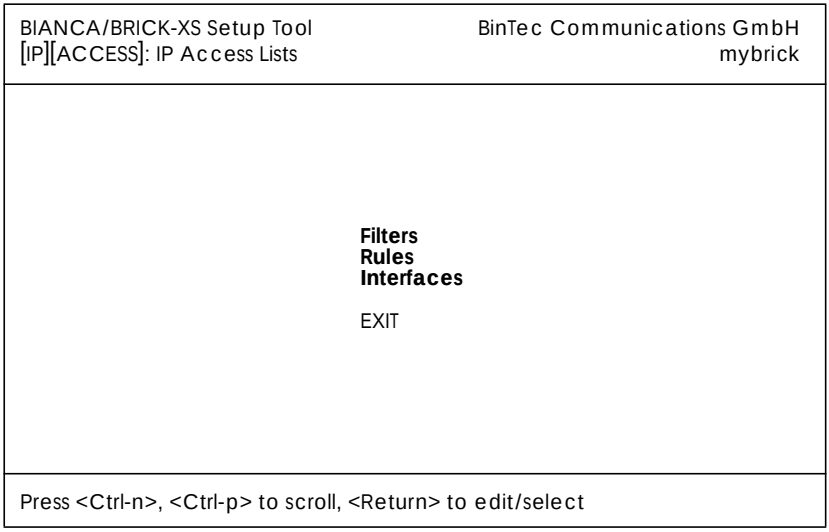
The diagram below shows in detail how Access List Rules and Filters are applied to incoming IP traffic.


Incoming IP Packets
ipExtIfRuleIndex = 0
yes
no
ipExtIfRuleIndex exist
yes
no
Discard IP packet!
Action = "ignore"
yes
no
Check ipRuleTable at ipExtIfRuleIndex
Does Filter exist?
no
yes
ipRuleAction
deny_if_not
allow_if_not
Does packet match referenced filter?
no
yes
ipRuleAction
deny
allow
else
else
Go to Rule at NextRuleIndex
Does this Rule exist ?
yes
no
Is this Rule 0 ?
yes
no
Route IP packet!

Setup Tool Menus

IP → ACCESS LISTS →

The IP–>Access Lists menu has changed and now displays three submenus where IP Access Lists settings are configured.



The FILTERS menu is used to configure filters. Each filter describes a subset of IP traffic and may be address, protocol, source or destination port based.

The RULES menu is used to configure rules. Rules can be ordered, or "chained" to control the order in which the filters are applied.

The INTERFACES menu is used to define which rule is used first for traffic arriving on that interface.

IP → ACCESS LISTS → FILTERS →

This menu lists the currently configured IP Access Filters and shows the Index number, Description, and Conditions for each filter. In the Conditions column abbreviations (explained in the menu) are used to describe the type of filter (i.e., address or port based filter).

To add a new filter select ADD. The menu shown below will be displayed.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[IP][ACCESS][FILTER][ADD]: Configure IP Access Filter                mybrick

    Description             no http
    Index                   4

    Protocol                any

    Source Address          192.168.50.5
    Source Mask             255.255.255.0
    Source Port             any

    Destination Address
    Destination Mask
    Destination Port        specify
    Specify Port            80

                  SAVE                    CANCEL

Enter integer range 0..65535
```

Description = A text string can be entered here to describe the filter. Note that in other menus only the first 15 characters of the description may be displayed.

Index = The index field can't be changed. The BRICK assigns a new filter number here automatically as new filters are added.

Protocol = Select a predefined protocol or "any" to match all protocols.

Source/Destination Address = (optional) Enter the source (or destination) IP address to match IP packets from.

Source/Destination Mask = (optional) Apply an optional mask.

Source/Destination Port = The range of port numbers to apply. Use “specify” to select a specific port number, “specify range” to select a range of port numbers by entering the first and the last port to be included in the range, “any” to match all ports numbers, or one of the predefined ranges, as explained in the table below.

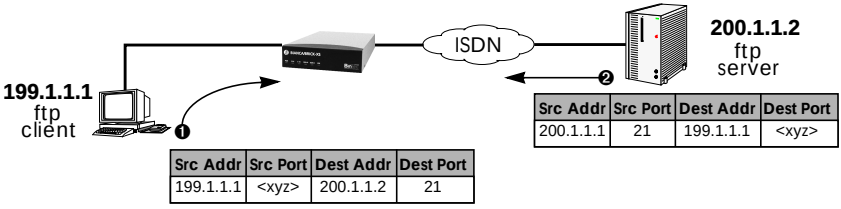
Source Port Ranges

0 ... 1023	1024 ... 4999	5000 ... 32767	32768 ... 65535
privileged	unprivileged		
server	clients	server	clients
specify / specify range			

Specify Port = If “specify” or “specify range” is set in the previous field the port number or port number range must be set here.

Using Source and Destination Port Numbers

Along with the source and destination addresses, the Internet Protocol uses source and destination ports numbers, to identify data connections uniquely. The client side generates a number (xyz) which is used as the source port, for the destination port it uses the number the server offers the service on. The server sends IP packets with the port numbers reversed in respect to the client. A simplified ftp connection might look like this.



IP → ACCESS LISTS → RULES →

This menu lists configured Rule Chains (individual chains are separated by a line). For each rule the Rule Index, Filter Index, Next Rule Index, Action, Filter, and Conditions are shown.

If a Rule (i.e., a link in the chain) is deleted from the list all neighbouring rules in the chain are automatically relinked.

Select ADD to create new rules. The menu below will be displayed. For each rule an Action and Filter must be defined that defines what to do when a packet matches that filter.

Select DELETE to remove an existing Rule that has been marked for deletion (Using the spacebar.).

Select REORG to reorganize the order of the rules in a chain. See the following page.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[IP][ACCESS][RULE][IP]: Configure IP Access Rules                mybrick

        Index
        Insert behind Rule    R2     F5     (no telnet)
        Action                deny M
        Filter                no ftp (1)

                        SAVE                      EXIT

Use <Space> to select
```

Index = This value can not be changed but is displayed when editing an existing rule. When creating new rules this field is empty until the rule is saved.

Insert behind Rule = (only shown when creating new rules) Use the scrollbar to select the location in the chain where this new rule should be inserted. For example: If you already have a global rule chain 1–3–2–0, selecting 3 here results in the chain 1–3–4–2–0.

To start a new (separate) rule chain use the scrollbar and select “none” in this field.

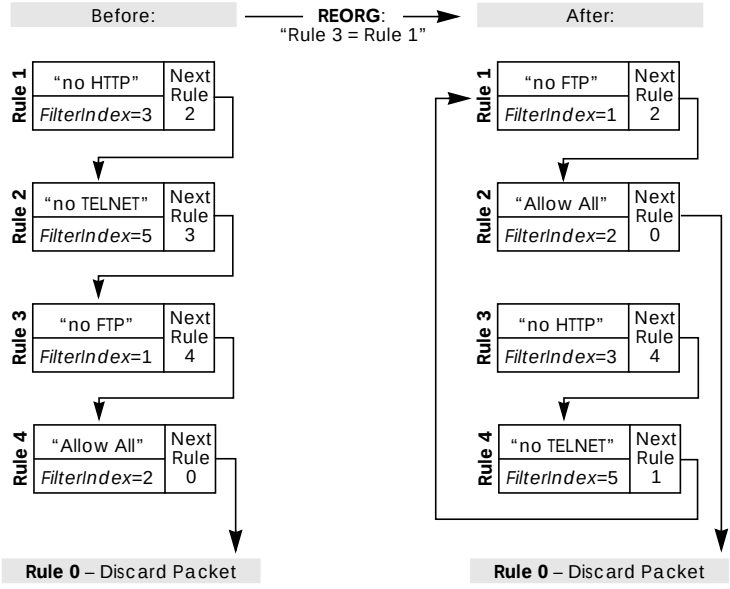
Action = The action field defines whether to allow or discard the packet based on whether or not the packet matches the filter (defined in the following field) or not.

Filter = The Filter to test IP packets against; use the spacebar to scroll through the list of currently configured filters.

Reorganizing Rules in a Chain

The REORG menu allows you to change the order of Rules in an Access Rule chain.

After selecting the Rule that should be placed at the beginning of the chain (the “Index of Rule that gets Index 1” field), remaining Rules are automatically relinked. The appropriate Rule Index and Next Rule Index numbers are reassigned in the ***ipRuleTable*** and the interface-specific Start Rules are updated in the ***ipExtIfTable***.



NOTE: The appropriate indicies are renumbered but the access semantics remain the same.

IP → ACCESS LISTS → INTERFACES →

This menu is used to control which Rule Chain(s) are used for packets arriving via the BRICK interface. This menu lists all IP capable interfaces and the First Rule that is currently being used for this interface.

To change the First Rule for any interface highlight the entry and hit Return key; otherwise select Exit to accept the displayed settings.

Note: By default Rule 1 is always used for newly created BRICK interfaces.

```
BIANCA/BRICK-XS Setup Tool                    BinTec Communications GmbH
[IP][ACCESS][INTERFACES]: Configure First Rules                 mybrick

   Configure first rules for interfaces

   Interface      First Rule              First Filter
   en1            0 (no access rules)
   en2            0 (no access rules)
   sales1         2                       3 (all else)
   sales2         2                       3 (all else)
   branch         2                       3 (all else)



   EXIT

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

In the EDIT / ADD menu the following fields are displayed.

Interface = This value can not be changed but is displayed for reference.

First Rule = Use the scrollbar to select the Rule to use first for packets arriving on this interface. Setting this field to “none” disables the Access List mechanism for this interface.

NOTE: If the referenced Rule doesn’t exist (in ***ipRuleTable***) then all packets arriving on this interface will be allowed.