

PBX PROTOCOLS

Some configuration values are automatically entered when a protocol is selected and saved. See the ip.buffer User Manual for additional information.

ASCII Lines

The default protocol will work in the majority of cases for both Serial and TCP/IP PBXs. Reads in single lines from the source and strips the top bit of data (D7), and strips control codes.

Option to force CR/LF line ending.

Alcatel TCP/IP [port 2533] (App Note: AN010)

The ip.buffer can collect CDR data from the Alcatel range of PBXs across the network.

Source = TCP Connect = “ipbuffer to Device (active/client)” *-forced*

Address = IP address of PBX Port = 2533 *-forced*

Note: It's vital that only one client connects to Alcatel at a time or CDR delivery is unreliable

Avaya RSP TCP/IP

The Avaya uses the Reliable Session Protocol (RSP).

Source = TCP Connect = “Device to ipbuffer (passive/server)” *-forced*

Allow = blank (or IP of PBX) Port = 9000 (default -can change, but must match PBX)

Set the Avaya: Enter the IP address of the ip.buffer

The Avaya can also be configured to deliver CDR by raw TCP/IP socket which is much more efficient. Use the standard “ASCII Lines” for this.

Binary (full 8-bit)

The binary protocol will store all incoming data and will not look for record terminators. All 8-bits of data are stored without any modification.

Generic Records

Designed for legacy PBXs, provides a more flexible protocol than “ASCII Lines”, because it will:

- terminate the record with CR+LF sequence
- discard incomplete records that do not terminate in the “Suffix” character string
- keep data between the Prefix and Suffix strings
- optionally send an Acknowledge string

Inter-Tel/Mitel Axxess & 5000 TCP/IP [port 4000]

Provides the sign-on process to initiate CDR records from the Inter-Tel PBX. The ip.buffer declares itself as “ip.buffer-00-02-ae-xx-xx-xx” (where xx-xx-xx is the serial number).

Source = TCP Connect = “ipbuffer to Device (active/client)”

Address = IP address of PBX Port = 4000 (or set to match PBX)

NEC NEAX (STX/ETX) Serial

This protocol is for the serial-connected NEC PBX where each chunk of data is enclosed in STX(0x02)...ETX(0x03) binary characters. Option for:

Text Lines Discards all other data, including the STX/ETX markers, and appends a CR+LF

Raw (STX..ETX) Saves all data, including the STX/ETX markers

ADDITIONAL PBXs

Cisco Call Manager (App Note: AN006)

The ip.buffer can collect from Cisco CCM5+ using FTP push. SFTP is not supported (in any case the ip.buffer should be located right next to the CCM to limit the possibility of data loss due to network failures, so packet sniffing will not be an issue).

Set ip.buffer: Source = FTP

Enter a suitable username and password into the ip.buffer

Set the CCM: Enter the IP address of the ip.buffer for one of the server destinations

Choose "FTP" (SFTP collection is not supported in the ip.buffer)

You can leave the directory blank, or enter "/".

Enter the username and password as set in the ip.buffer FTP collection page

Choose a frequency to push from the CCM

The CCM will make connection immediately to the ip.buffer when the settings have been saved. The connection stays open, and the CCM will push the data at the specified interval.

Note: You may want to use the "File Markers" feature. This inserts the {ftp...} markers into the file, since the CCM will push two separate files every time it does the push.

Cisco Call Manager Express (CCME)

The ip.buffer can collect from Cisco CCME using:

FTP Push (set as for the CCM above)

RADIUS Accounting (set Source =UDP, and port =1813)

Syslog collection (set Source =UDP, and port =514)

Siemens HiCom 300 (App Note AN003 - Connecting to Fixed IP Devices)

It has a fixed IP address, though later firmware allows it to be changed. The ip.buffer can connect:

- a) Directly to the Hicom300 using the ip.buffer multi-homing, or
- b) Via a standard router/firewall for absolute security and isolation.