2025

Mizu Softswitch Administrator's Guide

VoIP Server documentation

Mizu Server is a VoIP server application for the Microsoft Windows platforms

-

Mizutech
6/1/2025

Contents

1. Introduction

About

This document provides an overall technical description of Mizu VoIP platform. For non technical user guides please check out our website. Your installation may not include all modules described in this document and you will notice small changes if you program version doesn’t conform to the documented version!

Version

MizuServer v.10.2 Administrator’s Guide
Revisited June 1, 2025

Copyright

Disclaimer: MizuTech SRL reserves the right to change any information found in this document without any written notice to the user.

License Agreement and Trademark Acknowledgement

You must accept the license agreement (LicenseAgreement) before you use any Mizu hardware or software component.
The softswitch core is licensed by proprietary Mizutech license: All rights reserved for Mizutech.
Some of the extra modules are licensed by separate license terms. (These modules are included in unmodified binary format and integrated with the Softswitch via their API)
LINUX is a registered trademark of Linus Torvalds in the United States and other countries.
Windows and Microsoft SQL Server is a trademark of Microsoft Corporation, registered in the United States and other countries.
Oracle is a registered trademark of Oracle Corporation.
Most of the code have been written by us (Mizutech SRL) from scratch, but the software also uses the following open source code or binaries:

- G.711 codec: by Sun Microsystems Inc provided for “unrestricted use"
- G.729 codec: based on ITU-T G.729 Reference C code
- GSM codec: GNU Lesser General Public License version 2.1
- Speex and OPUS codec: by Xiph.Org Foundation license
- Freeswitch: MPL 1.1 license (some unmodified modules as-is)
- OpenSSL: Apache License
- OpenH323 (used in test tools) are licensed under MPL 1.0

Other logos, products, brand names and service names contained in this document are the property of their respective owners (trademarks or registered trademarks of their respective companies)

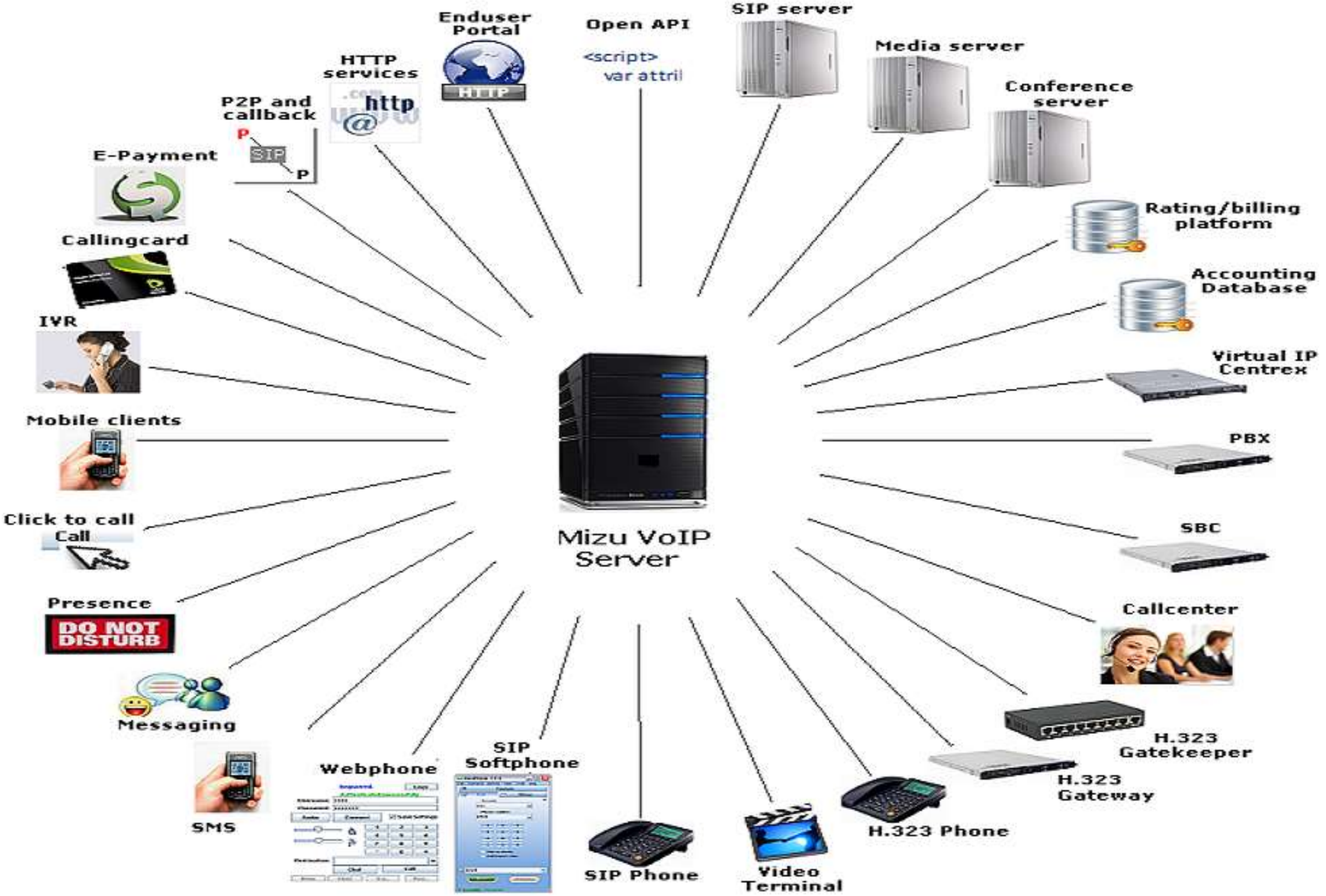
Introduction

This document describes the administration of Mizu Gateways, SoftSwitches and CallCenters. A unique set of proprietary software and hardware based capabilities and processes to build up a VoIP network, including planning and network management.

These components are designed to cover the telecommunication needs for small to very large companies and VoIP carriers. The main power of the system is the sophisticated VoIP components, which are strongly used in today's telecommunication infrastructures.

The Mizu components can be used as standalone or as centralized intelligent VoIP platform, capable to handle millions of minutes/month.

1.2. Features


Enduser Portal
HTTP
Open API
<script>
var attri
SIP server
Media server
Conference server
Rating/billing platform
Accounting Database
Virtual IP Centrex
PBX
SBC
Callcenter
H.323 Gatekeeper
H.323 Gateway
H.323 Phone
Video Terminal
SIP Phone
SIP Softphone
Webphone
SMS
Messaging
Presence
DO NOT DISTURB
Click to call
Call
Mobile clients
IVR
Callingcard
E-Payment
P2P and callback
P
SIP
P
HTTP services
http
Mizu VoIP Server

1.2.1. Hosting

- MS-SQL backend (Express or Full versions) or embedded database
- Ethernet 10/100/1000 Base-T
- Static IP, PPPoE (DSL or cable modem), DialUpISDN,VPN
- Encrypted communications
- Virtual servers
- STUN/ICE Support
- NAT Support
- Near-End and Far-End NAT traversal
- Multi-homed and multi-domain support

1.2.2. SIP

- Compliant with SIP rfc's
- UDP, TCP and TLS transports
- Proxy server
- Registrar server
- Location server
- Redirect server
- B2B routing
- PBX features
- Transcoding B2BUA
- Conference Server
- SBC (Session Border Controller)
- Routed and Direct voice
- Automatic NAT detection
- DID Direct Inward Dialing
- Voice Recording and Playback
- Absent Subscriber
- Abbreviated Dialing
- Multiple Subscriber Aliases
- Anonymous Call Rejection
- Access Control Lists
- Call Baring Incoming/Outgoing
- Toll Restriction
- Parallel Hunting
- Click 2 Call

- CLIP/CLIR
- DTMF generation
- Call-Forward on out-of-service
- Codec transcoding
- Advanced statistics support
- NAT traversal of signaling
- NAT traversal of media
- SIP Session timers
- RTP Timers and media timeout
- Blind SIP Registration
- Late Codec Negotiation
- Multiple SIP registrations per user account
- Can act as an SBC
- Max Session Setting
- Manage Presence
- Detailed call logs
- SIP/SIMPLE
- SIP Reinvites
- SIP-H.323 protocol conversion
- WebRTC-SIP protocol conversion
- Class 4 features
- Class 5 features

- RFC 2543 compatibility
- RFC 3261 compatibility
- RFC 2976 The SIP INFO Method
- RFC 3262 Reliability of Provisional Responses in Session Initiation
- RFC 2617 HTTP Authentication
- RFC 3263 Locating SIP Servers
- RFC 3265 Specific Event Notification
- RFC 3420 Internet Media Type message/sipfrag
- RFC 3515 Refer Method
- RFC 3311 UPDATE Method
- RFC 3581 Symmetric Response Routing
- RFC 3842 Message Summary and Message Waiting Indication Event Package
- RFC 3891 "Replaces" Header
- RFC 3325 Private Extensions to the Session Initiation

- RFC 2778 A Model for Presence and Instant Messaging
- RFC 3428 Session Initiation Protocol (SIP) Extension for Instant Messaging
- RFC 1889 RTP: A Transport for Real-Time Applications
- RFC 2190 RTP Payload Format for H.263 Video Streams -only routing
- RFC 2327 SDP: Session Description Protocol
- RFC 2833 RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals
- RFC 3264 An Offer/Answer Model with Session Description Protocol
- RFC 3550 RTP: A Transport Protocol for Real-Time Applications -replaces RFC 1889
- RFC 3555 MIME Type Registration of RTP Payload Formats
- RFC 3911 The SIP "Join" Header
- RFC 3324 Network Asserted Identity
- RFC 3326 The Reason Header Field
- RFC 3581 Symmetric Response Routing
- draft-ietf-mmusic-ice-02 A Methodology for NAT Traversal for Multimedia Session Establishment Protocols
- draft-ietf-avt-rtp-ilbc-04
- draft-ietf-sipping-cc-transfer Call Control - Transfer
- draft-ietf-sip-referredby-05
- Custom protocol extensions are possible

1.2.3. Codecs

- G.723.1
- G.729
- G.711 A-law
- G.711 u-law
- GSM 06.10
- GSM
- Speex 2,3,4,5,6 (narrowband, wideband and ultra-wideband)
- Opus
- G.726 (16,24,32,40 KHz)
- G.722
- T.38
- DTMF
- Custom 1 kbits codec
- All other codec’s for pass-trough
- Voice:
    - Adaptive de-jitter buffer
    - Voice Activity Detection/Silence Suppression

- Recording conversations (In Stereo caller/callee left/right)
- QoS
- Packet saver technology

1.2.4. IP Centrex

- Call Forward All/Busy/No Answer
- Caller ID
- RingGrouops
- Call Return
- Call Waiting
- Call Forking
- Call Hold/Retrieve
- Caller ID Block
- Selective Caller ID Blocking/Unblocking
- Speed Dial
- Direct Inward Dialing (DID)
- Three-Way Calling, Conference support
- Message Waiting Indicator
- Call transfer, Attended transfer, Unattended transfer
- IVR (all applications: call, callback, p2p, forward, etc)
- VoiceMail 2 Email
  DTMF transcoding on server side
- Interactive Voice Response (IVR) supporting applications such as credit card and prepaid services
- Video
- Conference calls
- T.38 fax relay
- SMS relay
- SMS commands (callback, P2P)
- Web interface

1.2.5. Call Center

- Automatic Call Distribution: like simple automatic dialing, power dialing, predictive dialing, predictive intelligent dialing
- IVR
- Call Recoding: All calls can be recorded and stored
- Real time call check out: Supervisors can listen to the ongoing calls real time

- PBX Features: Call hold, call wait, call transfer, call forward (conditional and unconditional), call conference, CLIP, CLIR
- Customizable Scripts: script tree, with any number of branches, answers, and reason codes.
- Customizable IVR: Any number of language, any number of branches, voice and faxmail, call transfer to the operators
- Statistic generation: customer statistics, operator statistics, call related statistics, work time statistics, campaign statistics
- Campaign creation: supervisors can create a campaigns
- Invitation letter: customization, and automatic printing
- Report generation: Specific hourly, daily and weekly reports

1.2.6. Accounting

- ACD Features
- Unlimited accounts / Unlimited Extensions
- Automatic pincode generation
- Flexibile authentication (digest,IP,port,user,etc)

1.2.7. Routing

- Custom Routing Rules
- Multi-Carrier Support
- ACL
- Sophisticated configurations
- Load Balancing on available devices
- Rerouting
- Number rewriting (calling and called)
- Failovering (multiple levels)
- Least Cost Routing
- BRS -quality based routing
- Call Control Features (Maximum Talk Time, Max Ring Time)
- Call routing based on PLMN tariff packages
- Blacklist/White list filtering
- Time of Day Routing
- Direct Inward Dialing (DID)
- Route capacity
- Outbound Dial Map
- Speed Dial Numbers
- Auto call forwarding
- ANI Routing
- IP Blacklists
- Custom VoIP Providers

- Fraud detection tools
- Support for NAT traversal
- Automatic capacity rebalancing

  Remote Linked Servers
- Automatic channel management
- Number portability support
- User authentication by username/password, IP address, techprefix, caller number
- GeoIP database
- Automatic SIM allocation:

  Sim allocation rules:

  Rules can be defined on multiple levels: global, partner, gateway, engine, simpacket, simcard, time

  -Static

  -will not modify gw settings

  -Limits

  -sl (day/month)

  -packet allowed intervals

  -min/max lines for partner

  -Priorities

  -sim partnerm, sim, gw

  -Desired

  -desired minute on packet

  -packet multiplier

  -Rotate

  -“minrotateival”, “desired”, “maxrotateival”

  -Price

  -min/max pricediff on obj, maxpricepermin for system/partner

  -Timetable

  -BRS

  -LCR

  -and many other options

1.2.8. Billing

- Flexible Rate Definition (peak/off-peak/flat/custom, end-user/provider/reseller/sales, etc)
- Automatic and Real Time billing (CDRs already includes the prices)
- Prepaid and Postpaid platforms
- Call Credit Limit Control
- Unlimited reseller accounts
- Callshops
- Directions (traffic sender,prefix,gateway,sim packet) and time based billing.

- Reporting and price comparisons (LCR)
- Balance and rating with SIP signaling or HTTP requests
- Invoice generation in different formats, PDF generation, email scheduler and invoice printing
- Complete call rating & accounting services for complex rating schemes
- Currency and VAT can be set for every packet. Time zone can be changed.
- Automatic online currency conversion
- Paypal and lot's of other payment gateways are supported

1.2.9. Calling Card

- Pin Generation Management
- Pin-less Number Registration
- Support for multiple account types
- Management of PINs generation, activation and deactivation
- Support for unlimited number of PINs
- Ability to deactivate accounts after certain period or date
- Import and export of PIN batches
- Management of call limit per PIN
- Routing restrictions
- Max call duration management
- Automatic User Generation

1.2.10. WebRTC

The Mizu VoIP server has full support for WebRTC since version 7.4 (including websocket SIP signaling and DTLS/SRTP encoding/decoding)

1.2.11. H323

- H.323 Standard Features (v.1,2,3,4)
  SIP-H.323 protocol conversion (Signaling and media when needed)
- Full H.323 proxy
- H.225.0 Call Signaling
- Fast Connect/Fast Start
- H.245
- H245 tunneling
- H245 in setup
- DTMF send/receive

- Watchdog
- Direct endpoint call signaling.
- Gatekeeper routed: call signaling (H.225.0).
- Gatekeeper routed: call signaling (H.225.0) and control channel (H.245)
- Gatekeeper routed: call signaling (H.225.0), control channel (H.245) and voice
- RTP Port Range (For firewalls)
- Child Gatekeeper capability
- Backup Gatekeeper capability
- Gatekeeper clustering support (neighbors, parent/child, alternates)

1.2.12. VoIP-GSM

Deprecated!
GSM components are not included with the default standard install and the required hardware is for extra costs.
Please contact us about the possibilities.

Hardware Components
No hardware components are required for H323 and SIP networks.

VoIP-GSM hardware listing:

- VoIP-GSM gateway
  - -8 channel gateway, best fit to any cheap DSL connection
  - -up to 64 simcard/gateway
  - -SIM server interworking capability
  - -Integrated antenna splitter
- SIM Server
  - -up to 750 simcard
- VoIP-GSM Server
  - -industrial PC
  - -fault tolerant
  - -server failovering capability
  - -distributed architecture
- Built in watchdogs to monitor the operation of the system components

GSM features

- Dual Band (900 / 1800 MHz or 850 / 1900 MHz)
- Half rate, full rate, enhanced full rate, SMS, USSD
- SIM server support

- Integrated antenna splitter
- 8 channels/box
- Up to 8 SIM cards per engine
- Multiple ways to handle incoming calls
- Call Forwarding
- Sending and receiving SMS messages
- Email To SMS Feature
- Inter gateway SIM routing
- SIM server interworking
- GSM cell selection and locking
- DTMF send/receive
- CLI restriction
- SIM Rerouting
- Locking to a given gsm cell
- Automatic SIM credit request and charge
- Voice Recording and Playback (In Stereo caller/callee left/right)
- SIM server interworking
- Virtual Channels

VoIP-GSM

- First centralized architecture for GSM termination
- Unlimited Gateway and TrafficSender support
- Multiple signaling protocol support
- Load distribution between the operational channels
- No hard limit on the number of simultaneous calls
- High availability
- High throughput (more than 50 million minutes/month)
- No additional Mizu hardware required
- Equipment management
- Channel management
- Simcard management
- Automatic recharge
- Access Control Lists
- Routing (see below)
- Billing (see below)
- Exploits almost any SIM tariff model
- Number translation

- Protocol encryption
- Media proxy
- Automatic time synchronizations
- H.323/SIP Gateway Topology Hiding
- Embedded firewall
- Enhanced Security (automatic detection of flood attacks)
- Web GUI for end-users
- Encrypted communications
- License management
- Distributed absolute fault tolerant system
- External system supervisor service (email and sms alerts, watchdog can restart failed subsystems)
- Can be used as virtual servers

1.2.13. Tunneling and encryption

Standard SIPS/TLS/SRTP are fully supported by default.

An additional tunneling and encryption module is built-in into the Mizutech VoIP server and you can enable from the global configuration or from the Config Wizard.
Details: https://www.mizu-voip.com/Software/VoIPTunnel.aspx

1.2.14. Push notifications

Push notification support is a built-in module enable by default. Configuration and integration is described here.

1.2.15. Management

- Centralized configuration and management for all software and hardware components
- MManage:
  - -easy to use, mdi style
  - -almost every data query is parameterized with traffic direction and time
  - -all data in one place
  - -lots of data can be obtained from sl,asr,acl forms
  - -global system analysis

- Create and edit network elements
- Remote maintenance of Mizu gateways
- Display of system information
- Real-time call status view
- Service restart functions
- Display of the current status of each gateway and channel
- Real time call supervision (with many grouping options)
- Real time channel supervision (with many grouping options)
- Statistics (Text based and graphical ASR,ACD,SL, etc) on any traffic direction and time scale
- Disconnect Reasons (with many grouping options)
- CDR monitoring, retrieval, direct CDR access
  Operator Console
- Global system analysis!
- Routing pattern selection
- Routing time selection
- Failovering (in case of channel, gateway, direction etc errors)
- Best Route Selection
- Billing module
- Balance module
- Real Time Capacity check
- Ability to insert queries directly into the database
- Blacklist filtering
- Self-analysis tools
- Detailed logging (multiple levels). Detailed call tracing capability
- Call simulations
- Reseller/Agent Registration and Management
- Capacity and system load reports
- Import/Export from/to any format (SQL, text, excel, etc)
- CRM Integration
- Restore and Backup

1.2.16. Limitations

- The maximum database size for basic gateways and servers is 4GB. If you need to work with more than 5 mil calls for more than 3 month, you should upgrade your license to the advanced version and use a proper MS-SQL server (not the Express editions). This doesn’t affect servers with less than 100 simultaneous calls. You can upgrade the SQL server/service anytime later (full compatibility).

1.2.17. Known issues

- Some class 5 features will work only with SIP protocol
- H323 GK doesn’t support username/password authentication (mostly used for call transit and not to provide enduser services over H323)
- RADIUS is compatible only with some servers
- Mizutech doesn’t provide support for client side (softphone) FAX and video debugging. (FAX and video are fully supported by the server and will work if you are using a softphone conform the SIP standards, however many software has limited interoperability or bugs)

For a more detailed feature list see the homepage and features pages.

1.3. Contact and tech support

Full remote administration can be done by mizutech support including continuous email support and 24/7 emergency phone support.
Contact us or visit http://www.mizu-voip.com for more details.
Email support: serversupport@mizu-voip.com

2. Modules

Depending on licensing, some modules may not be available in your release!

The Mizu Soft switch (Server) is the “brain” of the system. Depending on your needs, you can connect as many gateways as you want. The soft switch is built from several modules: sip stack, h323 stack, sip – h323 conversion module, media server, ACL, routing, billing, alerting. Almost all modules are installed by default and they can be enabled/disabled as needed. Key modules are listed below:

2.1.1. SIP Stack

The Mizu SIP stack was written in C++. It’s very fast and robust, currently used by voip service providers and carriers handling millions of minutes.

2.1.2. WebRTC

The WebRTC module will handle VoIP calls from modern browsers. Includes also a WebRTC – SIP bridge with media relay support.

2.1.3. H323 Stack

Capable to work as a simple Gateway or as a fully featured Gatekeeper.

2.1.4. SIP-H323 converter

Thank to this module, the protocol conversion is very transparent. You don’t even need to know if your partners use SIP or H323.

2.1.5. RTPM

RTPM is used with Flash clients to convert the signaling and media between Flash and SIP/RTP.

2.1.7. Media Server

If your server needs to route the media channels for many concurrent calls, you may need to use a separate media server, thus offloading the server traffic, and maximizing media throughput.

2.1.8. Routing

With the Mizu softswitch you can build very sophisticated routing scenarios. The routing is usually based on traffic direction and time. LCR and BRS routing are available.

2.1.9. Billing

The server will generate the detailed CDR after each call. Thus the billing can work nearly real-time. (Very important for prepaid systems). You can generate various reports and invoices based on a set of predefined rules.

2.1.10. Alerting and daily report

The server can send various reports and alerts based on predefined rules. The reports are sent by email or SMS.

2.1.11. Virtual Servers

You can create up to 100 virtual servers on a single pc. These are completely separate billing/routing/signaling entities.

2.1.12. Call Center

Manage operators, automatic call distribution, IVR and other callcenter specific tasks.

2.1.13. Watchdog service

This NT service can automatically detect critical service problems and restart the VoIP server, the SQL database or the OS.

2.1.14. API

The built-in HTTP server will listen for service requests.

2.1.15. Extra PBX

A separate module implementing a few extra pbx features such as voicemail and conference rooms.

2.1.16. Direct command interface

Administrative tasks and class5 service requests can be handled by the command line TCP interface.

2.1.17. Enduser web portal

Template portal is available with source code. All common end-user tasks can be done via a user friendly web interface (new user registration, download CDR records, show statistics, setup call forwarding, payment, etc)

2.1.18. Self-check and reporting

Various system thresholds will checked real-time and the server will do the corresponding actions automatically. The same module is responsible for daily/weekly/monthly email or SMS reports and alerts for administrators and users.

2.1.19. VoIP-GSM Gateway

Mizu VoIP-GSM Gateways support 8 concurrent calls and up to 64 simcards.
See the features section for more details.
GSM
All standard GSM capabilities are supported.
VoIP
Mizu VoIP-GSM Gateways can accept SIP and H323 registrations, can act as a SIP proxy or a H323 Gatekeeper or Gateway. These functions can be run simultaneously.
SIM Bank
The built-in simbank will allow to virtually routing the simcards in other Mizu gateways.
Mizu VoIP-GSM Gateways can take advantage of an external simbank, so you can have all your simcards in one place, easing the maintenance and administration tasks.

2.1.20. Other components

server service: the brain of the system
H323 GK: standard H323 gatekeeper
SIP Server: sip stack
HTTP interface: enduser/reseller website with built-in http server
Media Server: rtp routing
VGW: voip-gsm gateway, the most essential part of the client
SMS: sending short messages over HTTP API or SMPP
client service: this service supervises the gsm gateway and gives a clear interface to the server
MManage: smart client software, capable to manage the whole system
supervisor service: this service supervises the mserver
alerter service: collects statistical information and reports it
recplayer: can play g729, g723, encrypted, raw PCM and wave files
loganalizer: log file parser
gwtest: handle gsm terminal (no h323)

ipmux: packet saver client and server
serveremulator: server interface to gk
simalloctest: test the automatic sim allocation
smtp_test: test smtp functionality
tariffcalc: estimate sim packet real price
tcperver: tcp server for test
udptest: udp through test
valerterclient: alerter sw which can be installed on client computers
vchargecards: manage chargecards
vclientinterface: platform specific functions for the gw
partnerclient: admin sw. for our partners
pricesettings: for packet price configuration
routingandprices: for config. routes, prices and sim packet priorities
servertest: brute force test for the server
supervisor: supervises the server
updater: automatically updates client software from the server ftp
mediasrv: media server for routing rtp packets
businesslgc: controls the routing, registration, endpoint list, endpoint creation, udp initialization

VPC

Simple monitoring software for business purposes. Each partner (gateway or simcard owners, traffic senders, etc) can have their own VPC to monitorize their own traffic and create reports.

VPC Setup

You can give the VPC for any of your partners. The partners can login to the VPC with the username and password configured in the “Users and Devices” form in MManage. Usually only “Owner” users will receive VPC access.

You can define what users can see in their VPC by setting the “Can watch sim packets”, “Can watch users/devices” and “Access Rights” in the user configuration form (billing tab). See section 4.3.1 for more details.

The VPC included with MManage has the capability to login as a super user. To do so, you have to enter your partner username, but use the admin password (from the “ad” account). Then you have access to the “Add Query” button in the VPC. Here you can add,delete or modify the existing queries and their access rights.

In the “rights_allow” field you can put a list of user id, “all” or “nobody” fields. The same for the “rights_deny”. Thus you can configure which partner can see and execute which queries.

3. Tutorial

This VoIP server tutorial is a shorter comprehensive guide to cover the most important topics about the VoIP server usage.

The tutorial have been migrated into a separate document available from here:
https://www.mizu-voip.com/Portals/0/Files/mizu_voip_server_tutorial.pdf

4. Administration

This chapter discusses the administration details of the VoIP server.

4.1. MManage

4.1.1. Overview

All administration tasks can be done using the **MManage** (MizuManage) client application. The main module is automatically installed with the server installer, but you can also install it independently to any PC and manage your server remotely. Restricted administration is available also from the web interface. During this guide you will meet very often the "**global configuration**" term. This is referring for the settings configurable using the "Configurations" form (under the "Other" item). These settings are system global and are stored in the tb_settings database table (will be discussed later)

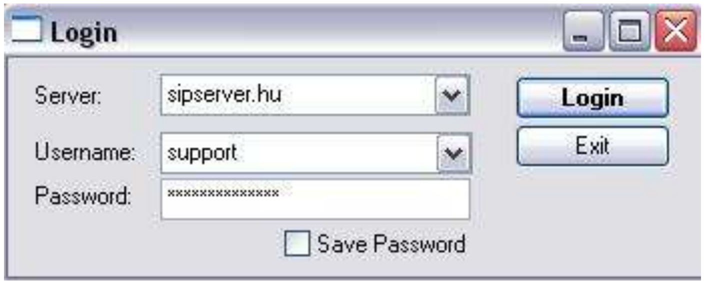
4.1.2. MManage Installation

The MManage program is part of the VoIP server or it can be downloaded separately from here.
The software is shipped as a standard windows install package. The requirements are the followings:
-Windows 2000/XP/Vista/Win7/8/10/11/2003/2008/2012/2014/2016/2019/2022/2025
-At least 1024x768 screen resolution for better operation
-You may need a headset for test calls
-Network connection
Double click on the install exe and follow the instructions.

During the install procedure the following **modules and files** will be copied:
-MManage.exe –the main executable
-Database connectivity tools
-VoIP client utility programs (SIP, WebRTC and H323)
-Utility programs: geoip, tariffcalc, smtptest, rptest, pdfcreator, sqlrouter, vpc etc
-Required dll files
-Help files
-Uninstall.exe
-Other files (depending of your install package configuration, OS version, etc)

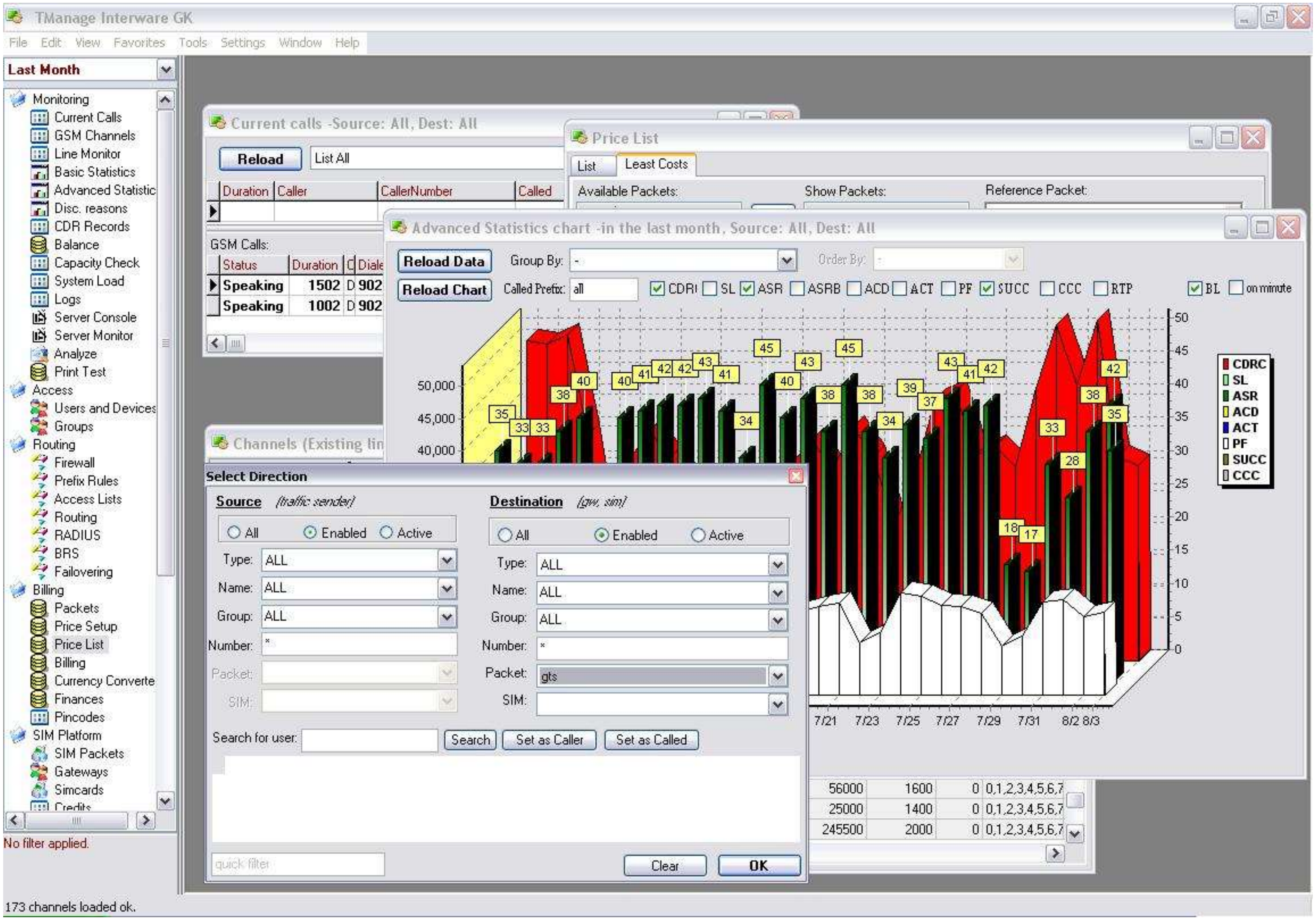
When properly installed, you are ready to login on your server(s) and/or your gateways. If you have a central server, all administration tasks can be done connecting only to the server. If your gateways are running without a server, you must connect to each gateway separately for doing administration tasks.

The following values are required on login:
Server: server address
Instance: application and database instance (a single server can hold several virtual server)
Username: login name
Password: login password



Server: sipserver.hu
Username: support
Password: **************
□ Save Password

4.1.3. MManage Framework



List All

Duration	Caller	CallerNumber	Called

GSM Calls:

Status	Duration	C	Diale
Speaking	1502	D	902
Speaking	1002	D	902

Group By: -

Order By: -

Called Prefix: all

☑ CDRI ☐ SL ☑ ASR ☐ ASRB ☐ ACD ☐ ACT ☐ PF ☑ SUCC ☐ CCC ☐ RTP ☑ BL ☐ on minute

Source: ○ All ● Enabled ○ Active

Type: ALL

Name: ALL

Group: ALL

Number: *

Packet:

SIM:

Destination: ○ All ● Enabled ○ Active

Type: ALL

Name: ALL

Group: ALL

Number: *

Packet: gts

SIM:

Search for user:

56000	1600	0	0,1,2,3,4,5,6,7
25000	1400	0	0,1,2,3,4,5,6,7
245500	2000	0	0,1,2,3,4,5,6,7

No filter applied.

173 channels loaded ok.

Almost all tasks are done by selecting an item from the left side of the main form. For detailed descriptions please read below.
In the Menu you can find common tasks such as “Settings”, “Save As”, etc. The selected action usually has effect only on the current active form.

From the **File Menu** you can save, print or export the selected form. Usual database operations are performed from the **Edit Menu**. In the **Favorites Menu** you can see the most frequently used items. In the **Config Menu** you can find a set of helper applications explained later in this document.

In the **Fields** the most important form is the "**Filter**" which will filter almost all listing used in MManage. Here you can define your preference regarding the traffic direction including Source and Destination. You can filter on Item Type, Item, Group, Number Prefix, Packet and SIM Card. *For example you may select one SIM ID, and when loading logs, you will see only the messages related to the selected simcard.*

In the left-bottom side of the form, you can find an edit box used for **quick search.** *You can use the '*' character in the begin and the end of expressions. (For example when searching for CDR records).*

Most of the report will be filtered after the selected **Date Interval** also.

In the Thresholds you can set some thresholds used for MManage. This setting doesn't have any effects on the server or gateway. Server and gateway thresholds may be set up from the Configurations Form explained below. In the **Options** Windows, you can set up several important MManage parameters.

In the **Help Menu** you have access to documentation.

In the **Licensing** box you can see your server parameters (there is no effect if you change these values, because these are used only for informing you). *Depending on licensing, some modules may not be available in your release!* Occasionally you may need to know the software version you use, which you can find in the **About** box.

Example: How the check your ASR for the traffic sender "A" in the last week.
1. In the date-time drop-down list, select the "Last Week" field
2. In the "Select Direction" form set the "Source" (left side) "Type" to traffic sender, and select "A" in the "Name" drop-down list (or type "A" manually)
3. Launch the "Basic Statisitcs" form under Monitoring.
4. Clear the "Group by" option (select the first "-" line)
5. Make sure the ASR checkbox is checked
6. Click on (Re)Load
7. Depending on current server config and current load this query may take some time (on a usual configuration this will take 2 second)

4.1.4. Import-Export Wizard

With the ease of the import –export wizard, accessible from MManage File menu, you can import and export data from/to a lots of data formats.
The following file types and databases are supported:

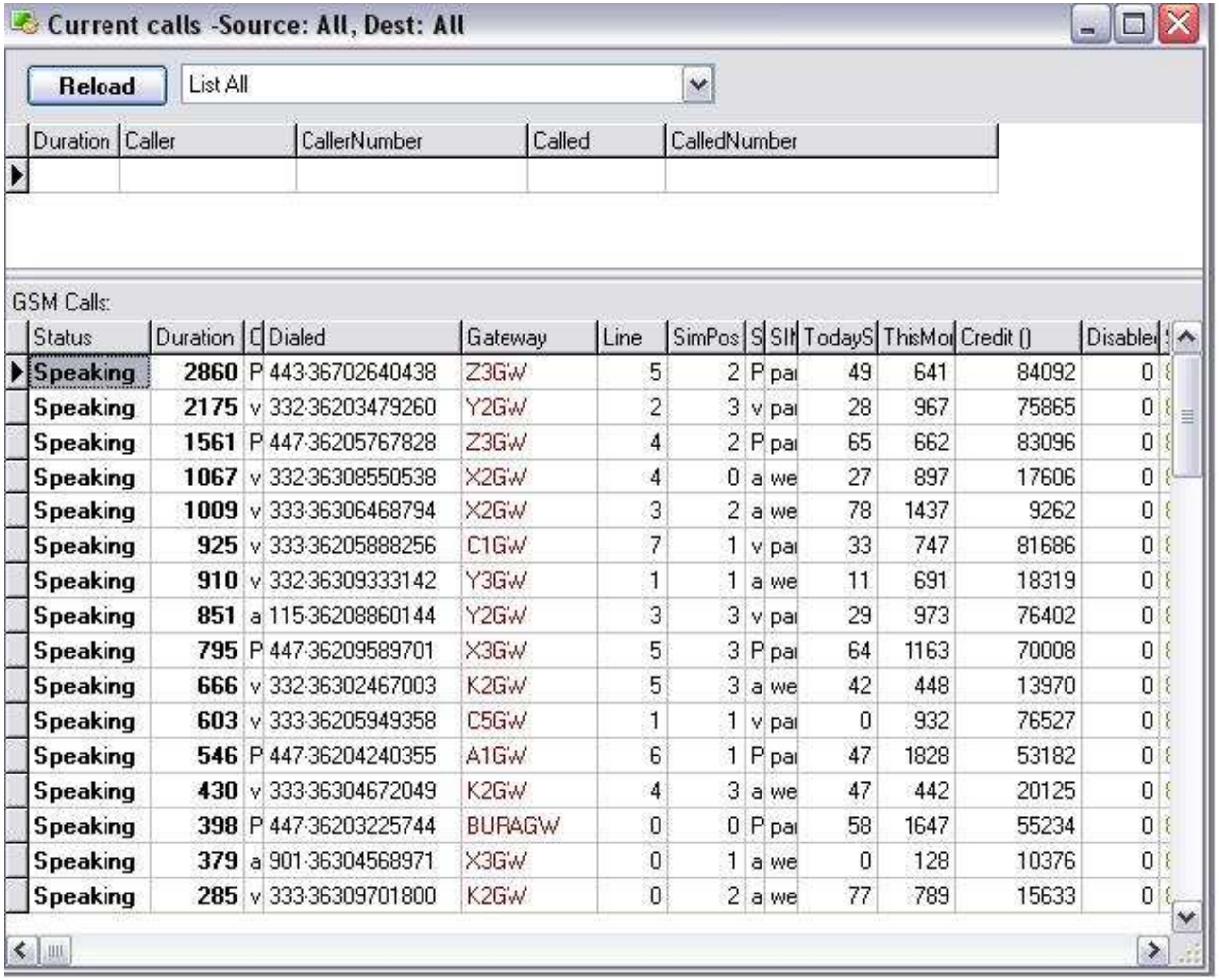
Access
Excel
DSN
CSV
Text

IBM-DB2
Interbase
MS-SQL
MySQL
Oracle
Paradox
DBF,dBase,FoxPro

Other data sources which can be accessed by ADO or ODBC.

4.2. Monitoring

4.2.1. Current Calls



Duration	Caller	CallerNumber	Called	CalledNumber

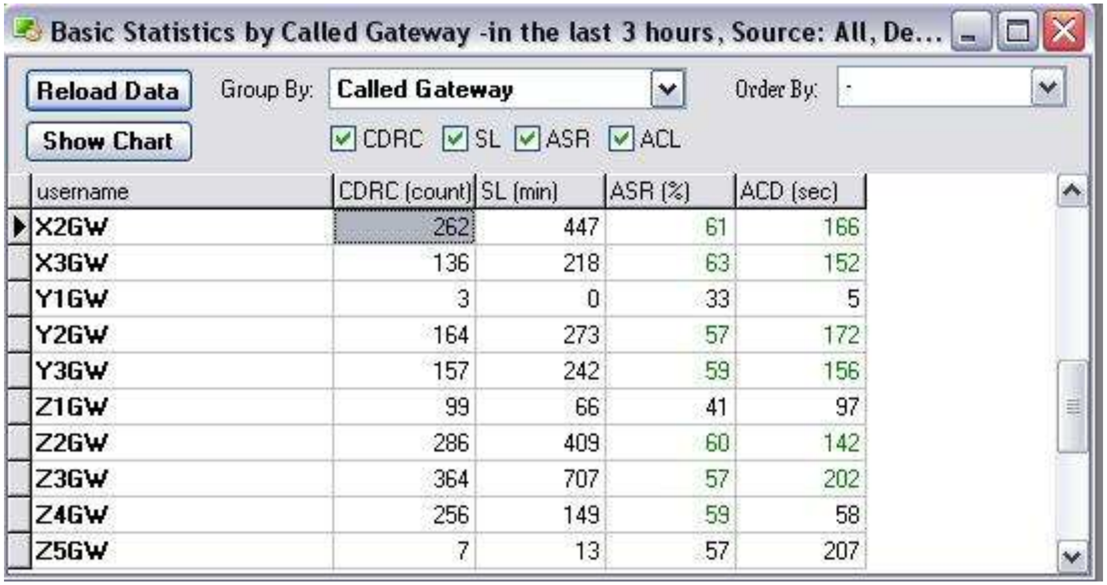
GSM Calls:

Status	Duration	C	Dialed	Gateway	Line	SimPos	S	SIl	TodayS	ThisMo	Credit ()	Disable
Speaking	**2860**	P	443·36702640438	Z3GW	5	2	P	pa	49	641	84092	0
Speaking	**2175**	v	332·36203479260	Y2GW	2	3	v	pa	28	967	75865	0
Speaking	**1561**	P	447·36205767828	Z3GW	4	2	P	pa	65	662	83096	0
Speaking	**1067**	v	332·36308550538	X2GW	4	0	a	we	27	897	17606	0
Speaking	**1009**	v	333·36306468794	X2GW	3	2	a	we	78	1437	9262	0
Speaking	**925**	v	333·36205888256	C1GW	7	1	v	pa	33	747	81686	0
Speaking	**910**	v	332·36309333142	Y3GW	1	1	a	we	11	691	18319	0
Speaking	**851**	a	115·36208860144	Y2GW	3	3	v	pa	29	973	76402	0
Speaking	**795**	P	447·36209589701	X3GW	5	3	P	pa	64	1163	70008	0
Speaking	**666**	v	332·36302467003	K2GW	5	3	a	we	42	448	13970	0
Speaking	**603**	v	333·36205949358	C5GW	1	1	v	pa	0	932	76527	0
Speaking	**546**	P	447·36204240355	A1GW	6	1	P	pa	47	1828	53182	0
Speaking	**430**	v	333·36304672049	K2GW	4	3	a	we	47	442	20125	0
Speaking	**398**	P	447·36203225744	BURAGW	0	0	P	pa	58	1647	55234	0
Speaking	**379**	a	901·36304568971	X3GW	0	1	a	we	0	128	10376	0
Speaking	**285**	v	333·36309701800	K2GW	0	2	a	we	77	789	15633	0

Currently running calls are listed here. Calls terminated on Mizu Gateways are displayed in separate list from other directions. You can filter the listing by selecting your preferences in the “Set Direction” box (as you can do in many other parts of the program).
The following grouping is available: by caller, by called, by called prefix, by simowner and by sim packet.
Field Explanations:
Status: engine or simcard status. Can have the following values: Gateway Disabled, Off (no info), Not Active, Gateway Disconnected, Closed, Not Ready, Ready, Dialing, Ringing, Speaking, Call Ending, DTMF, Simulating Outgoing, Simulated Incoming, Routing to SIMID, Routing to Alias, Routing
Duration: seconds elapsed from Setup (not from Connect!)
Caller: source name (user name or traffic sender name)
Called: destination name (user name or traffic sender name)
CallerNumber: the phone number of the caller party
CalledNumber: the phone number of the called party
Dialed: number routed to called user or gateway (with techprefix)
Line: the number of the gsm channel (usually from 0 to 7)
SimPos: the position of the active simslot in the current engine (usually from 0 to 7)
SIM Owner: the owner of the SIM Card
Packet: the type of the SIM Card
TodaySpeachLength: the number of active minutes on the current simcard since 00:00
ThisMonthSpeachLength: the number of active minutes on the current simcard since the first day of the current month
SIM ID: sim identification number

4.2.3. Basic Statistics

Shows the main traffic amount and quality parameters of your system.



Basic Statistics by Called Gateway -in the last 3 hours, Source: All, De...

Group By: Called Gateway

Order By: -

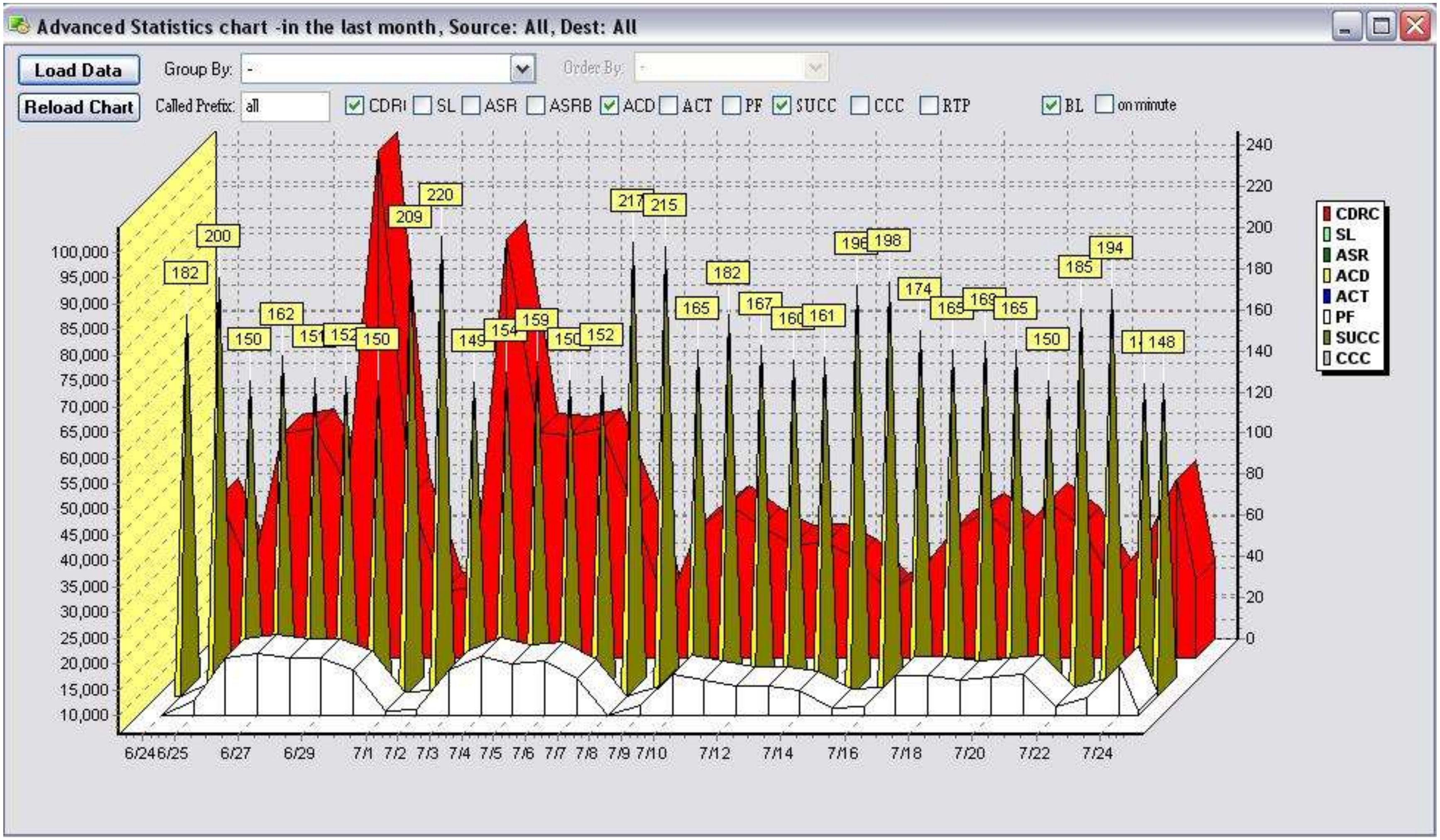
☑ CDRC ☑ SL ☑ ASR ☑ ACL

username	CDRC (count)	SL (min)	ASR (%)	ACD (sec)
X2GW	262	447	61	166
X3GW	136	218	63	152
Y1GW	3	0	33	5
Y2GW	164	273	57	172
Y3GW	157	242	59	156
Z1GW	99	66	41	97
Z2GW	286	409	60	142
Z3GW	364	707	57	202
Z4GW	256	149	59	58
Z5GW	7	13	57	207

CDRC: call attempt count
SL: speech length (duration in minutes)
ASR: average success ratio (percent)
ACL, ACD: average call length, average call duration (in second)
You can select any direction in the “Select Directions” Box, to check only that specific traffic. Also there are some simple groupings available:
-No grouping: will display the total sum. Chart views are supported only with this option
-Group by Called Gateway: list of destination gateway statistics
-Group by Traffic Sender: list of statistics by source
-Group by SIM Packet: statistics by SIMCard type
-Group by All caller and called
-Group by Provider Direction: statistics by called number prefix (first 4 digits)

4.2.4. Advanced Statistics

This is an extended version of Basic Statistics. You can find more grouping options here.
In some versions is shown simply as “Statistics”.



Advanced Statistics chart -in the last month, Source: All, Dest: All

Group By: -

Order By: -

Called Prefix: all

☑ CDRI ☐ SL ☐ ASR ☐ ASRB ☑ ACD ☐ ACT ☐ PF ☑ SUCC ☐ CCC ☐ RTP ☑ BL ☐ on minute

You can make reports based on the following fields:
-CDRC: number of calls
-SL: speech length (all call duration)
-ASR: the number of successfully answered calls divided by the total number of calls attempted (seizures) Since busy signals and other rejections by the called number count as call failures, the calculated ASR value can vary depending on user behavior
-ACD: average call duration

-ASRB: average success ration, but here the “success” means a minimum amount of duration. Configurable in Settings Menu -> Thresholds Box
-ACT: average connect time. The time elapsed from setup until the connect in seconds
-SUCC: successful call count (same as ASR but not in percent)
-CCC: concurrent (simultaneous) call count
-RTP: media channel statistics
-EC: end-user costs
-PC: provider costs
-SC: sales costs
-CC: company costs
-EP: estimated enduser price/minute
-PP: estimated provider price/minute
-PF: profit value (This require your billing module to be properly configured including provider prices)
-PR: profit margin in percent
-PFE: profit from enduser (useful if you are using reseller or sales accounts)
-PFR: profit from top reseller (useful if you are using reseller accounts)
-HM: hidden minutes

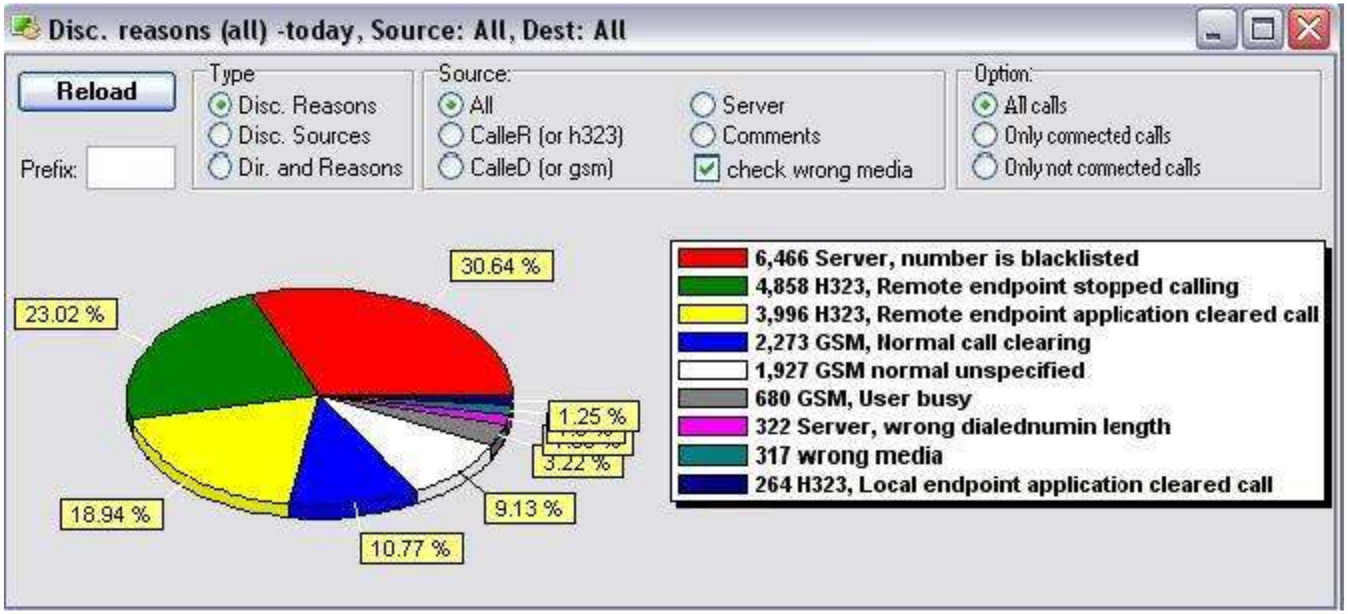
You can make the grouping by minute in this form by checking the “on minute” box.

The following “group by” options are available:
-: display summary data (no groupby)
Caller and Called: group by caller and called users
Caller: group by caller (source) user
Called: group by called (destination) user
Traffic Sender: group by caller (source) user, but show only traffic senders
Called Gateway: group by called (destination) user, but show only gsm gateways
GSM Engine: group by called gsm channels
Gateway, Packet and SIM Card: group by called simcard (and show the actual gateway and packet)
SIM Card: group by called simcard
Caller IP: group by caller ip address
Week –absolute: group by week, but with sum (don’t groupby to months)
Day –absolute : group by day, but with sum (don’t groupby to weeks)
Hour –absolute: group by hour, but with sum (don’t groupby to day)
Week: group by weeks
Day: group by days
Hour: group by hours
Minute: group by minutes
Day Compare: compare current weekday with last week the same day
Called SIM Packet: group by called simcards group

Partner/Day: group by partner and day
Partner/Hour: group by partner and hour
Partner/Minute: : group by partner and minute
Called Country: : group by called user country
Called Direction: : group by callednumber zone
Provider direction (prefix):: group by callednumber prefix
Provider direction (name): group by callednumber direction
Direction and packet: group by prefix and simpacket
Provider direction and packet: group by callednumber zone and simpacket
By caller root endusers: group by billed or company callerusers

4.2.5. Disc. Reasons

Disconnect codes in graphical form by any traffic direction.



Disc. reasons (all) -today, Source: All, Dest: All

Reload

Prefix:

Type:
● Disc. Reasons
○ Disc. Sources
○ Dir. and Reasons

Source:
● All
○ CalleR (or h323)
○ CalleD (or gsm)
○ Server
○ Comments
☑ check wrong media

Option:
● All calls
○ Only connected calls
○ Only not connected calls

🟥 6,466 Server, number is blacklisted
🟩 4,858 H323, Remote endpoint stopped calling
🟨 3,996 H323, Remote endpoint application cleared call
🟦 2,273 GSM, Normal call clearing
⬜ 1,927 GSM normal unspecified
▨ 680 GSM, User busy
🟪 322 Server, wrong dialednumin length
▨ 317 wrong media
■ 264 H323, Local endpoint application cleared call

The server will collect the reason in the most appropriate format depending on the protocols used. For example for a call from voip to gsm if the disconnect was caused by the gsm party, then you will see the GSM network reason code here. Otherwise, if the disconnect source was the caller party, and then you will see H323 or SIP reason codes here.

The most common reason codes are the followings:
-SIP, Bye: normal SIP close code
-SIP, CANCEL: the call was canceled by the caller (not connected call)
-H323, Remote endpoint application cleared call: normal H323 disconnect
-H323, Remote endpoint stopped calling: the call was canceled by the caller (not connected call)
-GSM, Normal call clearing: normal GSM close code
-GSM, Normal unspecified: normal GSM close code
-Server, Blacklisted: dropped due to ACL (blacklist)
-Wrong Media: no voice activity detected.

4.2.6. Line Monitor

This is a simple listing of your channels. You can discover all simcard problems by scrolling down this list. (Missing channels are highlighted)

4.2.7. Capacity Check

This module tests the capacity for the predefined direction in priority order.

4.2.8. System Load

Shows system utilization statistics such as concurrent calls, CPU load and RAM usage.
Note: AvM type means weighted utilization: average of max and average usage ((AVG(value) + MAX(value)) / 2)

4.2.9. Server Console

Direct interface to the server command port. Type help to see the available commands. You can connect directly to any gateway interface.
Command defined on gateway interface:

help	show this command list
info	show status and important parameters
cmd	launch the predefined process
exec	launch the predefined process
file	will send the requested file
showlog	will send the last lines from the requested file
timeset	will sent the current time
setini	write to config file
getini	read from config file
dtmf	send dtmf
ftpget	load from ftp
ftpput	put file to ftp

selfupgrade do a selfupgrade
gwrestart restart the gateway process
pcrestart restart the gateway (hardware)
and many more as described by the API (all API commands are accessible also from console)

Raw clear text CLI access is also listening on port configured by the adminport global config option.
CLI access might be IP filtered by the remoteadmin parameter. Possible values:

0: disable
1: from localhost/127.0.0.1
2: from same machine
3: from trusted sources
4: from local lan and subnet
5: from everywhere (default)

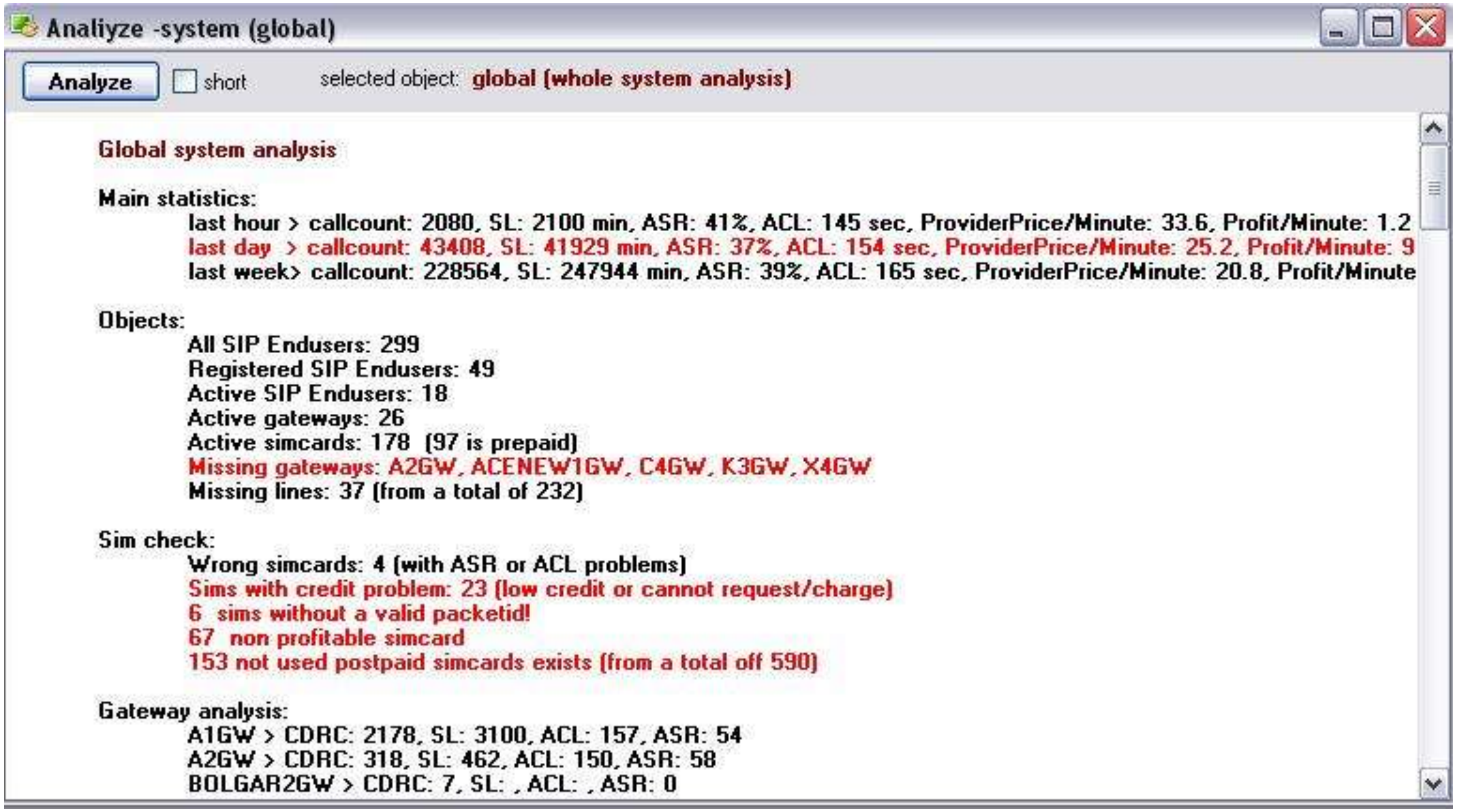
4.2.10. Server Monitor

Will connect to the server logport. The trace level depends on configuration (Open the Configuration form, type “log” in the filter box, and hit the enter button. Then you can see all options regarding to log levels)

4.2.11. Logs

Here you can see the log records for the server and every connected Mizu Gateways in the selected time period. You can restrict the listing by defining the source, severity or filtering.

4.1.12. Analyze



□ short

selected object: global (whole system analysis)

```
Global system analysis

Main statistics:
        last hour > callcount: 2080, SL: 2100 min, ASR: 41%, ACL: 145 sec, ProviderPrice/Minute: 33.6, Profit/Minute: 1.2
        last day  > callcount: 43408, SL: 41929 min, ASR: 37%, ACL: 154 sec, ProviderPrice/Minute: 25.2, Profit/Minute: 9
        last week> callcount: 228564, SL: 247944 min, ASR: 39%, ACL: 165 sec, ProviderPrice/Minute: 20.8, Profit/Minute

Objects:
        All SIP Endusers: 299
        Registered SIP Endusers: 49
        Active SIP Endusers: 18
        Active gateways: 26
        Active simcards: 178  (97 is prepaid)
        Missing gateways: A2GW, ACENEW1GW, C4GW, K3GW, X4GW
        Missing lines: 37 (from a total of 232)

Sim check:
        Wrong simcards: 4 (with ASR or ACL problems)
        Sims with credit problem: 23 (low credit or cannot request/charge)
        6  sims without a valid packetid!
        67  non profitable simcard
        153 not used postpaid simcards exists (from a total off 590)

Gateway analysis:
        A1GW > CDRC: 2178, SL: 3100, ACL: 157, ASR: 54
        A2GW > CDRC: 318, SL: 462, ACL: 150, ASR: 58
        BOLGAR2GW > CDRC: 7, SL: , ACL: , ASR: 0
```

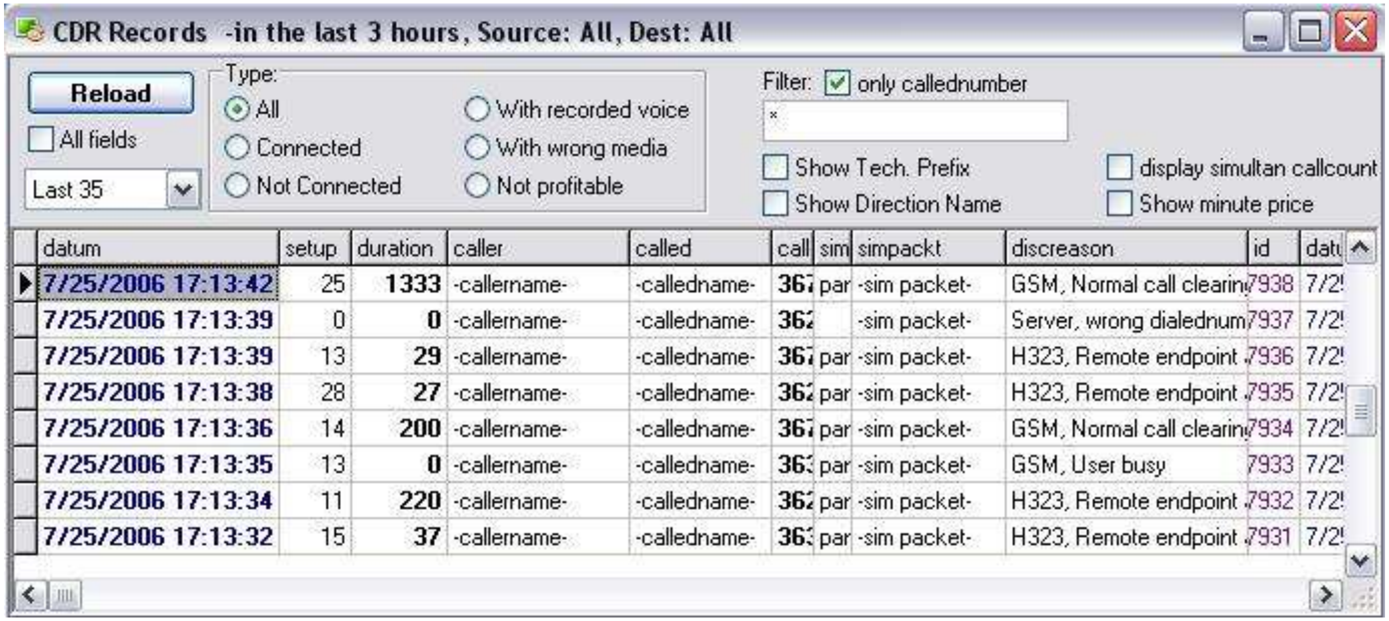
You will get detailed system analysis in this module. Thus you can see through the system by only one mouse click. Malfunctions are colored in red.

4.1.13. CDR

After every call (and SMS, etc), a new CDR (call detaild record) is stored in the database tb_cdrs table (and in tb_cdrresellers when the reseller option is used).
CDRs can be filtered, analyzed, exported and lots of vital statistics are based on this records.
CDRs will contain the following fields:
Id: database identifier. Auto increment
Datum: the date-time when the CDR were inserted into the database (call end time)
Callstartdate: call start time (first INVITE sent or received)

Callenddate: first disconnect code or CANCEL/BYE received or sent
Connectdate: first 200 OK received or ACK for 200 OK sent
Connecttime: time elapsed until call fail or call pickup (routing+ringing time)
Workenddate: used for callcenters and represents the time when the operator have finished to work with the current client (CRM updates, etc)
Realduration: speech length
Discparty: disconnect origination. 1=called or gsm, 2=caller or h323, 3=router (server)
Discreason: disconnect reason code. Explanations in tb_reasoncodes
Callerid: caller database id from tb_users
Callerip: the origination ip
Callernumber: caller phone number (or sip username)
Calledid: called database id from tb_users
Simid: called simid (if any)
Calledline: Engine (phone line) or the called proxy authorization id (from tb_proxyauth)
Calledip: the ip address of the called party
OrigCalledNumber: received called party number (not modified)
Callednumber: techprefix and the normalized called number. *If the server will block the call too early, than you may have the "origcallednumber" here (no techprefix and normalization)*
DialedNumber (calleddialed): the forwarded called number *(sometimes only the "callednumber" will be insterted here)*
Rtpsent: rtp packets from caller to called. 0 if no rtp routing. At least 1 if routed. If remains 1, then routing has failed
In case of sip this means rtp packets received from the called and sent to caller successfully
Rtprec: rtp packets from called to caller. 0 if no rtp routing. At least 1 if routed. If remains 1, then routing has failed
In case of sip this means rtp packets received from the caller and sent to called successfully
Rtplost: lost rtp packets
Rtpcodec: voice codec name
Rtpname: used for gateways
Rtpframes: rtp payload framed in one udp packet
Signalin: audio signal strength into the playback device
Signalout: audio signal strength received from the audio recorder device
Jittertime: used when jitter time is reported by gateways or softphones
Tpercek: hungarian specific. deprecated
Costprovider: call cost to the provider (ex. Tmobile)
Costenduser: cost for the caller in global currency (ex: a sipuser or traffic sender)
Costenduseru: cost for the caller in user currency
Costsales: sales commission if any
Costcompany: price for the reseller company
Costadditional (costother): used for reseller prices (in the main cdr)
Recfileid: if we have recorded the voice, then after this field we can found the recorded file

Mark (marker): for special CDR records: EMAIL (e-mail), SMS (sms), FAX (fax calls), FAIL (failovered), RER (rerouted), FWD (forwarded), TRANS (transferred), (conference), PRED (predictive) and to signal other important call types
Opworktime: used in callcenters to store the actual operator worktime
Opwaittime: used in callcenters to specify how much time the operator have been waiting for the current cal
Billingstep: loaded from price settings (endusercost packet)
Unitprice: loaded from price settings (endusercost packet)
Billingentry: loaded from price settings (endusercost packet)
Origduration: all original duration (because the “realduration” field can be modified on IVR 2 leg billings or when hidden charges are applied)
Resellerid: top reseller id in tb_cdrs. Actual reseller id in tb_cdrresellers.
Accessnumber: set when the call have been made trough a specified IVR access number
Origcallerid: used when the caller id have been modified during the call. For example the caller can be a “traffic sender” but after ANI or PIN authorization there is an enduser inpersonalisation
Alegduration: used for 2 leg calls (first calleg with ivr)
Blegduration: used for 2 leg calls (second calleg from ivr after callforward)
Comment: with details about the call setup and disconnect. Can contain a shortened message exchange log.
Dirid: direction name after the called number prefix


CDR Records -in the last 3 hours, Source: All, Dest: All

□ All fields
Last 35

Type:
● All
○ Connected
○ Not Connected
○ With recorded voice
○ With wrong media
○ Not profitable

Filter: ☑ only callednumber
*
□ Show Tech. Prefix
□ Show Direction Name
□ display simultan callcount
□ Show minute price

datum	setup	duration	caller	called	call	sim	simpackt	discreason	id	datu
7/25/2006 17:13:42	25	1333	-callername-	-calledname-	367	par	-sim packet-	GSM, Normal call clearin	7938	7/2
7/25/2006 17:13:39	0	0	-callername-	-calledname-	362		-sim packet-	Server, wrong dialednum	7937	7/2
7/25/2006 17:13:39	13	29	-callername-	-calledname-	367	par	-sim packet-	H323, Remote endpoint .	7936	7/2
7/25/2006 17:13:38	28	27	-callername-	-calledname-	362	par	-sim packet-	H323, Remote endpoint .	7935	7/2
7/25/2006 17:13:36	14	200	-callername-	-calledname-	367	par	-sim packet-	GSM, Normal call clearin	7934	7/2
7/25/2006 17:13:35	13	0	-callername-	-calledname-	363	par	-sim packet-	GSM, User busy	7933	7/2
7/25/2006 17:13:34	11	220	-callername-	-calledname-	362	par	-sim packet-	H323, Remote endpoint .	7932	7/2
7/25/2006 17:13:32	15	37	-callername-	-calledname-	363	par	-sim packet-	H323, Remote endpoint .	7931	7/2

Rtpsent and rtprec is 0 when media routing has failed (if we don't route the media, or the terminating endpoint don't send media info to us, the system will set there values to 1, so this condition will be true)

All prices in the cdr records are calculated with VAT included!

To find out the possibilities to list the CDR record for a user, please check this wiki.

4.3.14. Balance

Duration lists of several traffic types.

4.3.15. Callcenter Statistics

Statistics related to callcenter operations: Campaign and operator statistics.
StartTime: operator first login in MAgent in the current day
EndTime: operator last seen time in the current day
WorkTime: time when MAgent was running
ActiveTime: time when the operator is in "Automatic Call" form and not paused
OpWaitTime (WaitTime): the time elapsed from a new call request untill the first call connect. Smaller times represent more effectiveness. (the reason for the predective dialer is to reduce this time to minimum). The value will be stored for each connected cdr record.
OpWorkTime (PTime): The time elapsed from hangup untill new call request. This will represent the time spent by the operator for data postprocessing. Quick operators will have smaller opworktimes (but can be affected by the ammount of data to store) . The value will be stored for each connected cdr record.
TotalWorktime: MAgent runtime in that day
ActiveWorktime: When the automatic dialing form is active and not in paused
Called: number of called clients
Completed: clients marked as completed. Useful for meassuring the operators effectiveness.
Invited: clients marked as invited. Useful for meassuring the operators effectiveness.
Recalls: clients marked as need recall
CDRC,SL,ASR,ACL: traditional statistics. More details here.

4.3.16. Alerting

Admins and support users can receive email, sms and/or push notifications about important events such as critical problems, malfunctions and other server-side issues. Search for "sendalert" on the Configurations form for the related settings:

- sendalert: 0: no, 1: to support only, 2: filtered important errors only (DEFAULT), 3: unfiltered/all errors, 4: all
- sendalert_usr: send alert about user/trunk related issues. -1: auto (DEFAULT to sendalert_maxcount/3), 0: no, 1+: max per day
    - sendalert_usr_once: only one message per user. 0: no, 1: yes (DEFAULT)

- sendalert_srv: list of id's or username's to watch (of sip servers). alert will be sent on [not reachable AND sendalert_srv_minfailedcalls subsequent failed calls] OR [on sendalert_srv_maxfailedcalls]
    - sendalert_srv_minnoanswer: minimum number of no answer count. default is 5. set to 0 to disable
    - sendalert_srv_minfailedcalls: if set to 0, then will trigger alert on first unreachable, otherwise we must also have this number of failed call. default is 20
    - sendalert_srv_maxfailedcalls: def is 600. set to 0 to disable
    - sendalert_srv_mincallduration: minimum call duration in seconds to treat it as connected call. default is 20
    - sendalert_srv_mintime: min time in seconds from first failure (so the failed state must be at least this long before to trigger an alert)
- sendalert_email: 0: no, 1: yes (if email account is configured. otherwise will use cmailer for a while only) (DEFAULT)
- sendalert_sms: 0: no, 1: yes (if sms is configured) (DEFAULT)
- sendalert_push: list of push accounts (id's or username's)
    - sendalert_push_via: send push via our gateway if it doesn't have push capabilities. -1: auto (1), 0: this server only, 1: via mizu gateway if this server doesn't have push or fails, 2: via mizu gateway only, 3: via both local and mizu gateway
    - sendalert_mzpushallow: allow sendapush ip list (to be configured on mizu push server only at fcm.webvoipphone.com)
    - sendalert_mzpushblock: block sendapush ip list (to be configured on mizu push server only at fcm.webvoipphone.com)
- sendalert_to: 0: admin users only, 1: support users only, 2: both (DEFAULT)
- sendalert_maxcount: max message per day -1: auto (DEFAULT), 0+ exact value
- sendalert_delay: minimum time between alerts in minutes. default is -1 which will default to 30 minute
- sendalert_ignoreusrsetting: 0 to check tb_users.sendsmsalert and sendemailalert, 1 to ignore (DEFAULT)
- sendalert_mizusupport: -1 auto (DEFAULT), 0 to disable alerts to mizutech support user, 1 to enable

An email account should be configured for the server to be able to send the email notifications, otherwise the built-in cmailer@mizu-voip.com email account might be used for a while. An email account can be configured from the Config Wizard or by the emailhost, emailuser, emailpassword and other email related global settings from the Configurations form.
Push notifications can be sent to registered accounts with the with the ntype/TYPE parameter set to 3. Details here. If push is not configured on the server, then the notifications can be sent via the mizutech SIP PUSH proxy service.
The supervisor thresholds can be configured as described here.

The following commands can be sent from the Server Console for testing.

- Send push notification: sendpushu,3,from,to,msg
- Send push notification via mizutech proxy: sendpushmz,users,subject,msg
- Send alert notification: sendalert,subject,msg

4.3. Access

4.3.1. Users

This form (**Users and devices**) will allows to manage the users of the system (Endusers, SIP users, Administrators, Tech. Support users)
The most important user type is the **endusers**. These are usually registered on the server using the SIP protocol. They are allowed to call each-other usually for free (presence and messaging is also enabled by default).
For high amount of inbound traffic, you should use a **traffic sender** user instead of an enduser (and use IP authentication instead of username/password based)
For outbound routing you will have to add **SIP server** or H323 gateway user type and set the routing properly.
If you need both to accept and sent traffic to another server (carrier) then you have to add it both as traffic sender and SIP server.

Open the "Users and devices" form to see your users. There are a few template user created during setup. You can add new user by just cloning one existing user (select an user with the appropriate type, click on the "new user" button, the select "yes" when asked for cloning). This is useful if one of the many settings must have a predefined value, so you must set it only once, then if the new users all cloned the setting will be copied also). Otherwise you can create empty records and populate the required fields. Most of the fields are optional (with a preset default value) and only a few is very important which you have to change for each user like the username, password, credit, postpaid/prepaid, IP/AuthIP and the authentication type (needauth field).

Users can also be created in batch (Config menu -> Users)

The following types of users are defined in the Mizu VoIP server:
Enduser: represents retails customers or sip devices. Endusers are usually authenticated by SIP username/password and the account can be either prepaid or postpaid
Sub-enduser: Represents sip devices or callshop cabins. One enduser can have multiple sub-enduser account. This means that when a sub-enduser (device) makes a call, then the parent Enduser account is billed
Operator: used only for outbound callcenters and represent an Agent account
Calling-Card: this type of account can be used for inbound callcenter if you wish more separation between regural users and ivr users. Otherwise a regural enduser account can be also used for IVR authentication (calling card / PIN)
Reseller: resellers can manage their account trough the web interface and can create their own tarrifs. They can add other resellers or endusers below their account.
Owner: can be used to separate certain type of businesses. You can create one ore more owners and add all other users below the owner account
Traffic sender: from where you receive bigger amount of traffic (wholesale)
Sales: you can add sales people to the system and apply a specified rate as their discount
GSM GW: VoIP-GSM gateways
SIP Server: your outbound routes (carriers)
H323 GW/GK: any device (gateway or gatekeeper) using H323 protocol
ISDN GW: gateways to the PSTN network

SMS GW: sms gateways usually with HTTP API
Support: support people will receive automatic maintenance related emails
Admin: admin people with special permissions (for example you need to login as an admin user to the web enduser interface to be able to have access to the customization options)

The "Users and devices" form can be filtered in many ways:

- quick filter field
- Fields menu -> Filter
- user type
- the drop down list from the user form
- listings initiated from other forms
- groups
- For more flexibility you might use direct SQL queries against the "tb_users" table.

You can **list** the users with the following filters:
-ActiveNow: gateways with received status in the last 5 minute or endusers active (register or invite received) in the last 3 hour
-Active:
 -gateways with received status in the last 24 hour or when "mustbeactive" is set to 1
 -endusers active (register or invite received) in the last 24 hour
-All Enabled: where the "Enabled" field is not 0
-All: all users
-New Users: users added in the last month
-New Web Registrations: users added in the last month by the web registration form
-Low Credits: will list users with credit lower than 3000

The properties can be modified from one of the following **pages**:

- Listing page (grid). All fields are listed here with add/delete/edit possibilities. The "monitor" field are updated based on current statistics and settings.
- Edit operator: showed only for callcenter operators to set the agent related settings
- Edit: the most important user settings are listed here
- Functions: change class5 features from here
- Billing: billing related options
- Gateway configuration: only used for linked gateways
- Show details: will list most of the important settings
- Analyze: view user statistics

The following **fields** are defined:
ID: database id. Auto increment

Type: user type (together with the “isoperator” field)

0: enduser (usually a sip user). -isoperator field set to 2 (default)
0: sub-enduser-isoperator set to 0
0: Operator (callcenter agent or callshop agent) -isoperator set to 1
0: Calling card –isoperator set to 6
1: reseller (usually a sip reseller)
4: sim,gw or traffic owner (sim partnerid or gateway parentid show this id)
5: traffic_sender (parentid can be a simowner or a gatewayowner)
6: sales (parentid is the reseller id)
8: gsm gw (parentid is the gatewayowner)
9: outbound sipserver (proxy or gateway) (parentid is the gatewayowner)
10: h323 gw (parentid is the gatewayowner)
11: isdn gw (parentid is the gatewayowner)
12: sms, fax,email gateway
14: support users (can operate with MManage, has ftp account)
15: admin users (can see and modify everything, can receive email/sms reports)
17: other users (for various reasons)

ParentID:

This field is very important for resellers/subresellers relationships.

Logical parent of the user. Checked for routing pattern, max lines, etc.

if a sipenduser then reseller company
if traffic sender, then traffic owner
if gateway, then gateway owner
if reseller company, than operator

BillingUserID:

Where the invoices will be sent. Credit will checked for these users

If the current type is an end-user, then can have a BillingUserID where we send the invoices. If not set or the same ID as the current, than the bills will be generated to itself. For example in a company, all bills will be sent to the boss (company address), nit the employee

IsBilledUser: set to 1 if this user is not a real service user, but a user who pays for other user. Usually this is a company who pays for its employee.

Deprecated!

UserGroup: users can call each other only if the user group is the same (default: 0)

usually users with the same parentid

RingGroup: a list of usernames separated by comma (all number will ring when the actual user will be called)

BelongsToCompany: when a company has more then one subscriber. Used for example for short sip numbers.

IsCompany: if the current user actually is a company

RGMode: how to use the ringgroup: 0=forked call, 1=round robin

Name: user first and last –name

Country: sip phone country (important for prefix rules)
ContactName: additional name
UseCallingCard: if has calling card (usable with pin codes)
CanDial: example: for sipuser is 1. for simowners is 0
Phone: user phone number (but not the sip phone)
Email: where the user can be contacted
Address: where the user can be contacted
Billaddress: where to send the invoices
TelNumber: sip telnumber.users can be contacted if we call there username or telnumber.
More than one DID or ANI number can be assigned for one user. For this the tb_users_othernumbers table is used where the type field is 0 for DID numbers and 1 for ANI numbers.
ShortTelnumber: sip short telnumber (for example if several users has the same BelongsToCompany field)
DisplayName: how the user will be displayed. Can be null
Username: the most important field. Used for authentification and also as a DID number. This field is unique and cannot be empty.
Password: password applicable everywhere (sip, web, VPC, etc)
Ip: sipphone, sipproxy or gsmgateway ip address. The server will overwrite with the last known ip address
AuthIp: if we want to authenticate after ip, not after username/password
More than one auth ip or domain can be used for a traffic sender. For this the tb_users_authip table is used.
NeedAuth:

-If NeedAuth is 0, then the system is an open voip relay !!!!
-If NeedAuth is 1, then AuthIP must match (usually from SIP traffic senders)
-If NeedAuth is 2, then TechPrefix must match *(usually from H323)*
-If NeedAuth is 3, then TechPrefix and IP must match (usually from H323)
-If NeedAuth is 4, then user/pwd must match (usually from SIP end-users)
-If NeedAuth is 5, then username must match
-If NeedAuth is 6, then AuthIP and Port must match
-If NeedAuth is 7, then AuthIP and username must match
-If NeedAuth is 8, then AuthIP and username/password must match
-If NeedAuth is 9 then AuthIP Range must match
-If NeedAuth is 10, then AuthIP Range and username/password must match

AddDate: when the user has been inserted in the database
Rights: rights on user interfaces

0: no access
10: cannot login (disabled)
20: can login but no rights
25: restricted user (for example support under reseller)
30: a normal user

40: sales
50: admin
60: general admin

AddedBy: the user id who have added this user (sales, web registration, etc)

Commission: used for sales to define their commission percent from the enduser price

Reduction: sales user can give to enduser some percent (subtracted from their commission)

LateFee: applicable when the user is late paying the invoice cost

PacketID: billing for users, traffic senders

BillingDay: usually 1 (the first day in every month)

Qualification: the importance for the user. From 0 to 10. for example if the user has big priority, then we route its calls to better routes

Postpaid: if the user will prepaid or postpaid

PaymentMode: Check (0), Bank Tranfer (1), Cash (2), Else (3)

ContractNumber: contract for end-users

Allowedpartners:

Allowed traffic senders for the gateway, or allowed gateways for traffic senders.

A list of user id separated by comma or '*'

Note that parent users will be checked too

Enabledprefixes: can be one prefix (with any length) or a list of prefixes with 3,4 or 5 digit separated by comma.

Can be used for trafic senders and gateways too. No need to setup a separate routing pattern if you use this restriction.

EnabledTechPrefixes: enabled techprefixes for the specified gateway (3 digit length numbers separated by comma)

BlockPrefixes: list of called prefixes that will be blocked for the user (techprefix will not be considered here). Numbers listed here must have 7 digit length and separated with comma.

ContractState: the status of the contract

0- Unknown
1- Not applicable
2 -In Progress
3 -Active
4 -Terminated

ContractComment: additional comment for sales

Credit: when postpaid, then we also can set a max amount (which will reset in every month)

Enabled: if disabled, it behaves as if it were deleted

DomainName: sipproxy domain name

Port: signaling port

TransIp: secondary signaling ip

TransPort: secondary signaling port

RouteRtpCaller: routing mode if this endpoint is the caller

0=check called settings –this is the preferred settings

1=don't touch the sdp and the rtp

2=sdp correction if necessary
3=route rtp if both behind nat
4= route rtp if caller is behind nat
5= route rtp if called is behind nat
6= route rtp if any endpoint is behind nat
7=always route rtp

RouteRtpCalled: routing mode if this endpoint is the called

0=check caller settings
1=don't touch the sdp and the rtp
2=sdp correction if necessary
3= route rtp if both behind nat
4= route rtp if caller is behind nat
5= route rtp if called is behind nat
6= route rtp if any endpoint is behind nat
7=always route rtp

If the routertp setting is set to 1 or 7, it will be applied regardless of the other endpoint setting.

RtpIp: last rtp ip
RtpPort: last rtp port
ServerRtpPort: last bind (we try to use the same for every user)
NatDetected: 0= no and don't change, 1=no but can be changed, 2=yes but can be changed, 3 yes, and don't change it
NatDetectDisabled: deprecated
Status: 0=inactive,1=registered, 2=speaking (if statusdate is too old, then treat as 0)
StatusDate: last status change
CalledNumber: last called number
CalledID: last called id
Discount1: discount percent. users can have discounts in for max 3 directions
Direction1: prefix. users can have discounts in for max 3 directions
Discount2: discount percent. users can have discounts in for max 3 directions
Direction2: prefix. users can have discounts in for max 3 directions
Discount3: discount percent. users can have discounts in for max 3 directions
Direction3: prefix. users can have discounts in for max 3 directions
TechPrefix:

The server can authorize and/or route the traffic after the incoming techprefix.
Sip users can have techprefixes too. this is usually common for reseller company users.
If no techprefix is specified, then it will be loaded from tb_pxrules if any.
Sim owners and vpc users can have a list of prefixes separated by comma.
If no techprefix is specified, 111 will be inserted for incoming called numbers.

If the techprefix is „-1”, then the original techprefix will be forwarded.
If the techprefix is „-2”, then the original techprefix will be inserted in cdr record (but not forwarded).
If the techprefix is empty, then only the normalized callednumber will be forwarded.
The following techprefixes are reserved for the server: 111,222,999.
Only 3 digit techprefix is allowed. If your traffic sender needs another techprefix length, you must rewrite the incoming number in the “Prefix Rules” form.
Example: protcoll: sip, Type: ip, value: your traffic sender ip, rewritefrom: oldtechprefix, rewriteto: newtechprefix.

Addtechprefix: we insert this number before the callednumber if the caller don’t send its calls with tech prefix
MaxLines: max concurrent calls allowed
maxlinetouse: deprecated
CurrCallCount: current running calls (usable for traffic senders)
enablefakegw: if we don’t have capacity, we can route h323 calls to a fake gateway to prevent congestions
candisablesim: if the router will check the disableduntil field from tb_sims
alarmat: we can ring the sipuser if it is set
forwardonbusy: telnumber where we have to forward the calls when busy
forwardonnoanswer: telnumber where we have to forward the calls when we have no answer (or * if to forward to “best” child user)
forwardalways: rerouting
voicemail: if we can send messages as email
mincreditonroute: if user has less credit, then we don’t even route the call
regtimeout: reregistration interval for sip proxies
maxsubsfail: we set the „nopriority” field when we reach „maxsubsfail” failed calls
subsfails: successive calls with duration smaller than 20 sec
nopriority: this gateway has lowered priority in the routing until this date
noprioritycount: successive lowered priority countminasr:
minasr: minimum asr before failover
minacl: minimum acl before failover
mincallcount: min. Cdr records to calculate minasr and minacl
lastrouted: last call time (applies to caller devices and users or to callers when it is a proxy)
lastcalled: last call time (applies to called devices and users)
active: applicable for gsm gateways.
display: text to display instead of username
description: important comment
comment: any comment
lastrectime: last status receive from this gsm gateway
realgw: we can have fake voip-gsm gateways
temporarilydisabled: gsmgw is temporarily disabled
onlytestcalls: we allow only calls with techprefix 999
testprefix: we allow only this techprefix

datum: when the user has been inserted into the database
mustbeactive: if the gsm gateway must be active. Will do actions if this field is 1 and the gateway is not active
notactivecount: how many time we found that the gw is not active
channelcount: gsm channel count
minline: minimum active lines. If we found less line active, then we do actions
nominlinecount: : how many time we found that the gw has not enough line
prioritypartner: this partner will have priority on this gateway
callerpriority: this caller prefix will have priority on this gateway
calledpriority: this called prefix will have priority on this gateway
autopriority: set by server. If the gateway is wrong, then we lower the priority until this time
absolutepriority: if we set it greater then for other gateways, all calls will be routed here, until it is filled, regardless to other routing settings
priority: gateway priority
swversion: gateway sw version
lastrestart: gateway last restart
cutg711: if a better codec exists for the caller (g723 or g729) than PCMU and PCMA will be not offered to the called party
pingtime: deprecated
avgkbitssec: deprecated
maxkbitssec: deprecated
bandwidth: deprecated
restartcount: gsm sw restart count
pcrestartcount: gsm gw (pc) restart count
lasterror: last error message from this gw
lastlog: last log message from this gw
sendonlyrec: where to send sip messages.

0 = received address and the address in the signaling (via,contact,etc)
1=send only to the source address
2=send only to address specified in the signaling

callsigaddr: h323 port
isfake: we can have fake voip-gsm gateways
forwardearlystart: if we can send media parameters before callstart (OK for INVITE). 2 if check called
changesptoring: if we have to change the session in progress message to ring. . 2 if check called
identityforward: we can toward these kinds of usernames and the other we rewrite to „identityrewrite”
identityrewrite: if the caller username don’t match the identityforward prefix, then we rewrite it
PlayAdv: if we can play advertisements for this user
Maxmonthlycredit: max allowed credit/month even if the user is postpaid (in ft not in filler)
Maxmonthlycreditend: max Maxmonthlycredit (because we increase Maxmonthlycredit by maxmonthlycreditinc every month if the user was active)
onlylocalaccess: traffic sender traffic will not be forwarded (can call only local users from tb_users and tb_numbers)

Maxmonthlycreditinc: determines how much money we add to Maxmonthlycredit every month
ContractNumber:
Contact Status:

```
0-Unknown
1-Not applicable
2-In Progress
3-Active
4-Terminated
```

Contract comment: any usefully comment for sales here
Noanswertimeout: will redirect if no answer received
Denyaddr: because the server will try to send the sip messages to all possible addresses, sometime it will missroute. With this setting you can restrict the address posibilities. Check the FAQ for more details.

sendfakealert: used for gsm gateways. Specifies the timeout in sec after that the gsm gateway will send an alert to voip even if no ringing have been received from gsm network. Set to 0 or -1 to disable. Gsm gateway settings will overwrite the traffic sender settings if is not set to -1
sendsmsalert: use for support and admin accounts. Will send sms notification to the configured “phone” when a critical error occurs
sendemailalert: use for support and admin accounts. Will send email notification to the configured “email” when a critical error occurs
sendsmsreport: send daily sms report
senddailyemail: send daily email report. 0: no, 1: yes, 2: include also CDR’s, 3: save only to file
sendmonthlyemail: send monthly email report. 0: no, 1: yes, 2: include also CDR’s, 3: save only to file
convertdtmf: applicable for sipproxy users

-0: DTMF messages will be forwarded as received from client
-1: INFO and RTP DTMF messages will be converted to InBand
-2: INFO and RTP DTMF messages will be forwarded and converted to InBand too.
-9: don’t forward INFO

Missed by SMS: notify about missed calls by sms. Usually used for endusers
Missed by Email: notify about missed calls by email. Usually used for endusers

Can watch sim packets: list of packetid separated by comma, used for VPC access. The actual partner can see this simpackets with his VPC account
Can watch users/devices: list of users and gateways id separated by comma, used for VPC access. The actual partner can see this devices with his VPC account
Access Rights: specify which fields are allowed for the user in the VPC application

0: simcard and traffic sender fields are not shown
1: simcard related fields are not shown (simid, packetname)
2: traffic sender related fields are not shown (name, username)
3: all fields are shown

CLI: CLIR and CLIP settings

0: forward always (forward asserted as normal number always!). will not hide, even if caller was set so
1: normal handling (forward asserted as normal number) -default
2: forward as asserted identity always (identityrewrite asserted)
3: forward as asserted identity only to trusted domains (identityrewrite asserted)
4: normal hide (no idenityrewrite forwarding)
5: force hide (no asserted identity too!). always hided

IsOperator: specify if the user is a callcenter operator (1) or a sub-enduser (0) or power enduser (2)

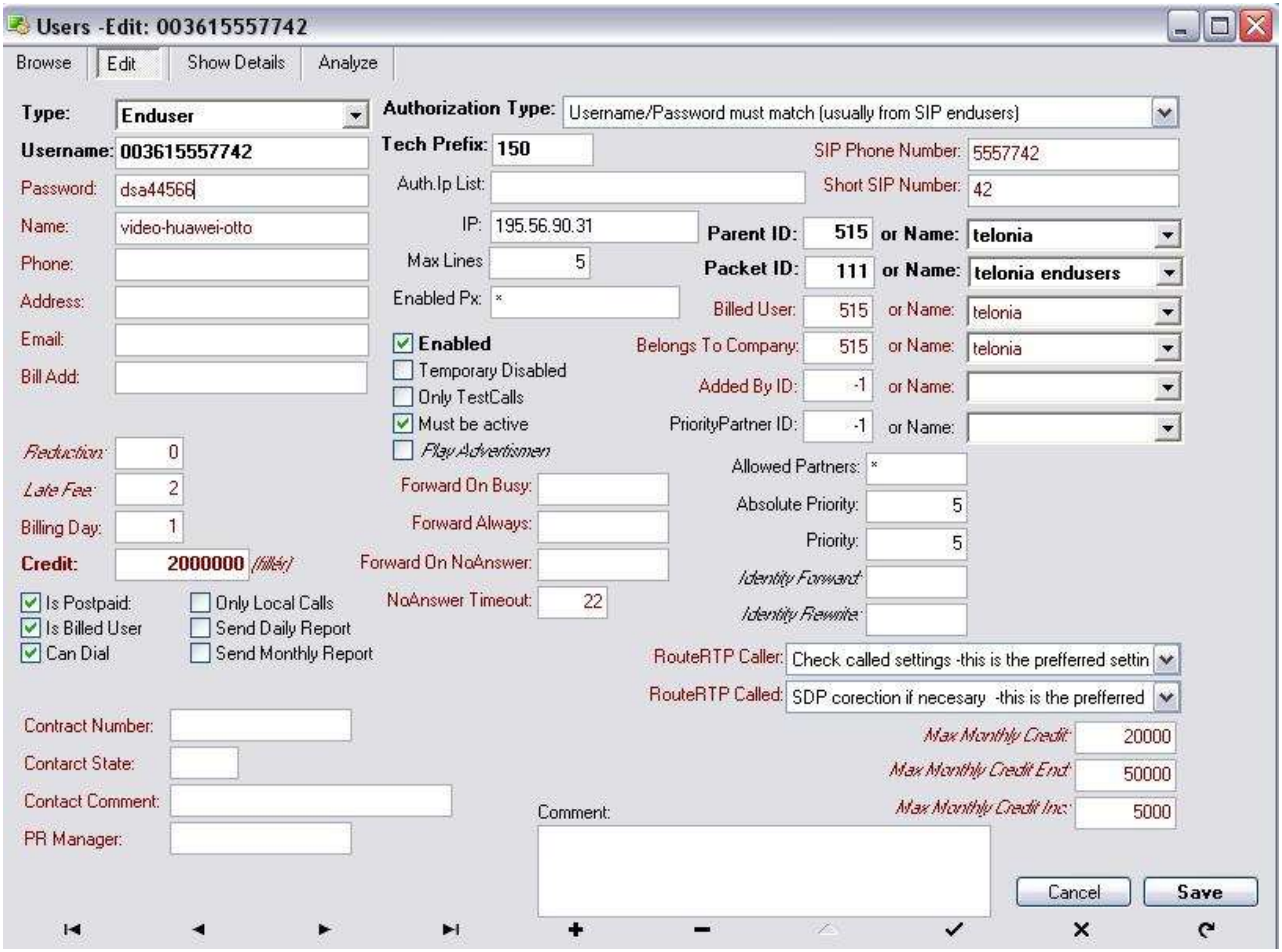
Choosecodecs: list of supported rtp payload formats in priority order separated by comma. Only one will be selected. Don't set this field to disable selecting only one code.

If set, than only one codec will be left in the sdp (plus the dtmf codecs). This will help, when the server answer to invitation with more than one codec in the 200. The client should answer with the final codec in the ACK, but many endpoint fail to do so.

SessionTimer: use session keep alive.

- 0: don't use
- 1: load from global config
- 2: autodetect (and using the sesskeepalive interval from the global configuration)
- 3: force
- other: use with the specified timeout (minutes)

For example if we set it to 5, than a UPDATE or reINVITE will be sent in every 5 minute to the other party. Please note, that the session keep-alive is not the same with NAT keep alive (which is used with every endpoint automatically)



Users -Edit: 003615557742

Type: Enduser

Username: 003615557742

Password: dsa44566

Name: video-huawei-otto

Phone:

Address:

Email:

Bill Add:

Reduction: 0

Late Fee: 2

Billing Day: 1

Credit: 2000000 *(filler)*

☑ Is Postpaid:
☑ Is Billed User
☑ Can Dial
□ Only Local Calls
□ Send Daily Report
□ Send Monthly Report

Contract Number:

Contarct State:

Contact Comment:

PR Manager:

Authorization Type: Username/Password must match (usually from SIP endusers)

Tech Prefix: 150

Auth.Ip List:

IP: 195.56.90.31

Max Lines 5

Enabled Px: *

☑ Enabled
□ Temporary Disabled
□ Only TestCalls
☑ Must be active
□ *Play Advertismen*

Forward On Busy:

Forward Always:

Forward On NoAnswer:

NoAnswer Timeout: 22

SIP Phone Number: 5557742

Short SIP Number: 42

Parent ID: 515 or Name: telonia

Packet ID: 111 or Name: telonia endusers

Billed User: 515 or Name: telonia

Belongs To Company: 515 or Name: telonia

Added By ID: -1 or Name:

PriorityPartner ID: -1 or Name:

Allowed Partners: *

Absolute Priority: 5

Priority: 5

Identity Forward:

Identity Rewrite:

RouteRTP Caller: Check called settings -this is the prefferred settin

RouteRTP Called: SDP corection if necesary -this is the prefferred

Max Monthly Credit: 20000

Max Monthly Credit End: 50000

Max Monthly Credit Inc: 5000

Comment:

Default users:
Owner mycompany: template for owner

GsmGw LOCAL_GW: used for advanced gateways. IP=127.0.0.1
GsmGw NOGW: used when no route found. IP=1.2.3.4, isfake=1,realgw=0
GsmGw FBACKUPGW: used to handle traffic exceed. IP=127.0.0.1, callsigaddr=1725,isfake=1,realgw=0
SipProxy: sip2h323: convert signaling form h323 to sip and from sip to h323. TechPrefix=-1,IP: 127.0.0.1?, Port?
TrafficSender: virtserver X: for routing traffic from virtual servers. AuthIPList: 192.168.0.1, TechPrefix=XXX
Enduser: PREDECTIVE_DIALER: IP: 4.3.2.1, TechPrefix?, username=cc_callbacknumber

The following fileds has direct impact to routing:
Type
ParentID
RingGroup
UserGroup
onlylocalaccess
enabledtechprefixes
accessrights
enabledprefixes
blockprefixes
enabledroutes
RGMode
TechPrefix
forward...
temporarydisabled
onlytestcalls
testprefix
prioritypartner
allowedpartners
callerpriority
calledpriority
absolutepriority

4.3.2. Devices

Administration of Mizu Gateways, Other GSM Gateways, H323 Endpoints, SIP Proxies, ISDN Gateways and other compatible devices. The fields are the same as for the Users (see above)

If the actual sip proxy require authentification, then we store the accounts in this table

Id: database identifier. Auto increment
Priority: Account priority (accounts will be used in priority order or in round-robin if they have equal priority)
Username: sip username used in authentification

Password: sip password used in authentification
CallerNumber: usually the same as username. If left as blank, then the server use the actual caller username.
Credit: account balance. When it reach 0, then we switch to the next account if any
DateEntered: record insertion date
LastUsed: the date-time when the server was routed some calls with this account
ProxyID: to which proxy the account belongs
Enabled: set to 0 to disable the usage of this account
SubsFails: the number of subsequent wrong calls with this account. If subsfails will reach a predefined value (30 as default), it means that there is some problem with this account or the money/time limit have been expired, and the server will switch to the next account if any

4.3.3. Groups

Grouping of several items will ease the administrations tasks. The following type of items can be grouped:
SIM Packets
Users
Gateways
Traffic Senders
These groups then can be used to simplify your routing and billing.

4.3.4. Ownership

Useful when you have arranged your users in certain hierarchies. For example reseller chain relationship.
Resellers can have an unlimited child-parent relationship (limited by the ”maxresellers” global config options).
To define the relationships we use the tb_users id and parented fields.

4.3.5. User authorization

Registration and authentication

User authentication can be done multiple ways (AuthType field in the user table).
Registration and authorization answers are cached in the Mizu server, so for subsequent requests from the same ip:port doesn’t have to query the database again. This means that if you change the password in the database it may take some time until it is considered.
Mizu server has built-in DOS attach protection. This means (among others) that after a few unsuccessful registration (wrong password) request from a UA that will be banned for a time. This banned list can be cleared from the server console with the “delbanned” command. Even whole IP ranges can be banned.

For example if there are too much meaningless or not authenticated request from an IP address (probably the attacker), than that IP address will be banned for a time period and the incoming messages from that IP will be silently dropped.

Users and devices will be allowed to access the system (and create new dialogs) only if they pass basic authorization which can be set from MManage -> Users and Devices -> Settings page.
For IVR access the server will authenticate the actual end-user based on the A number or will request a PIN code.

Basic authorization

Dialog authentication can be performed in the following ways:
-Open Relay: if you set the NeedAuth to 0 for a user, then your server becomes an open relay (this is forbidden by the “enforcestrongauth” global config by default)
-Authentication based on IP address: for this you have to set NeedAuth to 1 and enter the peer IP address in the AuthIp field (can be a list of ip address separated by comma). Instead of IP address you can also use a domain names here.
-Authentication based on tech prefix: this is mainly used in h323 network. Set the NeedAuth to 2 and enter a valid techprefix for the user (which is usually a traffic sender)
-IP and techprefix: NeedAuth must be set to 3. The “TechPrefix” and “AuthIp” fields must be set correctly
-Username/password authentication: usually for your sip endusers. NeedAuth must be set to 4. Username and password fields must be accordingly
-Authentication based on username: A number authentication. NeedAuth must be set to 5 and with a valid username
-IP and port based authentication: gives you better security than just IP authentication and also it is useful when you have more traffic sender from the same domain. NeedAuth must be set to 6. Port and IP have to be set accordingly. (port is stored in the callsigaddr field in tb_users. You need to edit it if needed)
-Username and IP: both username and authip must match

**SIP endusers are usually authenticated based on username and password.*
**Traffic senders (carriers) are usually authenticated based on IP address.*

Access numbers

Access numbers are special users. You will have to create them like usual users but their ivrid have to be set to a valid campaign id. (this is then linked with an IVR script)

For callback access you also need to set the “iscallback” user field properly. Read the “callback” services for more details.

A number authentication

For calls from traffic senders the server can try to authenticate the caller as an enduser if the “anumberhandling” option is 2,3 or 4 (by default it is set to 3).
More details in the IVR authentication section below.

IVR authentication

For IVR calls the server will do a “callingcardauth” global config option based authentication.

Please note that in this case the caller device is already authenticated based on basic authorization settings. The IVR needs to find an enduser to allow further operation, like call forward.

The server can authenticate the user based on the following methods:

- ANI/CLI authentication: if the CLI is known and this method is allowed.

A number authentication can be used to try user authentication for a call coming from a traffic sender. If user is found with the actual A number then the caller will be authenticated as the enduser, otherwise will be authenticated as traffic sender. In this case you can require a PIN number from the user.
The A numbers are usually extracted from the "From" field of the incoming SIP INVITE.
Don't enable A number authentication if you don't trust the traffic sender or there is a possibility that the A numbers are not received correctly.

Configuration options:
anumberhandling global configuration

- 0=disabled
- 1=only add
- 2= only accept
- 3=add and accept (default)
- 4=only a number access (no pincode request)

anumberhandlingforrouting global configuration

- 0=disabled
- 1=disabled
- 2=yes, but autoturnoff if not used for a while
- 3=always yes

enableanumberlookup per user configuration. You need to set it to 1 for traffic senders to enable or 0 to disable.

A numbers can be also registered by the users on a web interface or by sending an SMS in the proper predefined format.

- PIN (calling-card) based authentification:

When the A number is not known or the A number based authentication is disabled, the IVR have to ask the user for a valid PIN code.
This can be done by the CallingCardAuthentication IVR action.

After the server collects the DTMF digits it will lookup the database for a valid user entry. The authentication can be based on username, password or username+password or depending on the "**callingcardauth**" global config option which can have the following values:

- 0=all combinations with minlength check (default)
- 1=callingcard username
- 2= callingcard password

- 3= callingcard username+password
- 4=any username
- 5=any password
- 6=any username+password
- 7=any username+password or username+pin
- 8=username for callingcard and username+password or username+pin for other users (default)
- 9=pin
- 10=password or pin
- 11=all combinations
- 12= all combinations with minlength check (default)

When using the pin field based authentication, make sure that user has valid pin codes set in this field (when the users are automatically generated, the pin is set to be username+password).

User rights

User rights can be further restricted by several configuration option.
The most useful tool for this is the **routing** table. You can define were a certain user or a user group can initiate calls.

The following restrictions can be applied per user:
Allowedusers: list of the users or groups (prefixed with 'g') that is allowed to call the user. This can be used to restrict the access to an access number for example.

AllowedPartners: comma separated list of allowed partners and traffic senders. '*' will allow all. You may restrict the access on gateway or simpacket level instead of setting it for all simcards separately. Try to use the packet "allowedpartners" setting and leave it as '*' for the simcards!

Enabledprefixes: can be one prefix (with any length) or a list of prefixes with 3,4 or 5 digit separated by comma.
Can be used for traffic senders and gateways too. No need to setup a separate routing pattern if you use this restriction.
EnabledTechPrefixes: enabled techprefixes for the specified gateway (3 digit length numbers separated by comma)
BlockPrefixes: list of called prefixes that will be blocked for the user (techprefix will not be considered here). Numbers listed here must have 7 digit length and separated with comma.

MaxLines: max concurrent calls allowed (separate value for peak or offpeak)

Maxmonthlycredit: max allowed credit/month even if the user is postpaid

Maxmonthlycreditend: max Maxmonthlycredit (because we increase Maxmonthlycredit by maxmonthlycreditinc every month if the user was active)

Onlylocalaccess: traffic sender traffic will not be forwarded (can call only local users from tb_users and tb_numbers)
Maxmonthlycreditinc: determines how much money we add to Maxmonthlycredit every month

Access Rights: specify wich fields are allowed for the user in the VPC application

- 0: simcard and traffic sender fields are not shown
- 1: simcard related fields are not shown (simid, packetname)
- 2: traffic sender related fields are not shown (name, username)
- 3: all fields are shown

Global configuration options:
MAXSPEACHLEN: max allowed call duration in sec
allowedusers: max ring time in sec
callmaxwait: max waittime allowed for operators between calls (for administrative purposes)
enforcestrongauth: enforce authorization and strong passwords

More rate limit settings can be found in the “security” and “billing” related sections.
Try to avoid prioritizations by users, gateways, simpackets or channels (absolutepriority, priority, allowedpartners, prioritypartners, etc)
Almost all kind of configuration can be set up by using only the “routing” form.

LDAP, Radius or authentication via external database or API can be also added.

4.3.6. DID numbers

DID (Direct inward dialing) is a feature for routable static/dynamic numbers. Endusers can be reached directly by calling DID numbers from outside networks and DID numbers can be also used as a CLI (display number/A number) for outgoing calls. It might be used to share a SIP trunk among one or multiple users.

DID numbers can be acquired from carriers (for example the carrier where you are otherwise send the VoIP traffic), specialized DID providers such as didx: https://www.didx.net/ or other sources. These numbers can be assigned manually or via API. (integration needed with your did provider for this functionality). The numbers then can be stored in the tb_telnumbers table (Phone numbers form in MManage) and then manually or automatically assigned to users. (If enabled, on new user signup the user can select a DID number if any).

In the mizu servers you can manage DID on multiple levels.

- ➢ **Assign one DID per user**

This is the simplest and more robust usage, recommended if you have enough DID’s to assign to all your (business) users.
Just set the DID as user Username.

If for some reason you don't wish to use DID as username (for example if DID is assigned later) then you can use the TelNumber field to assign it to the user. The server will automatically accept calls either to username/othernumber/telnumber/shortnumber. For outgoing call it will also guess the best one to use from these fields (or it can be set explicitly by the replacecalleroncalls global config option)

- **User sign-up from the web portal**

You can configure the enduser web control panel so the users can get a DID number during sign-up.
Use the following configuration options:

  - autogeneratenewuseraccounts: 0=no,1=auto generate random,3=auto select from tb_telnumbers,6=user select from tb_telnumbers
  - autogeneratedid: 0=no,1=auto generate random,3=auto select from tb_telnumbers,6=user select from tb_telnumbers
  - allowenduserdidedit: allow enduser to edit its own telnumber field 0=no,1=yes for resellers,2=yes for endusers

- **Assign multiple DID per user**

You can use the othernumber list for this (tb_users_othernumbers). The same DID can be even assigned for multiple users. In this case the incoming call is routed to the "best" user (Best effort: user online, not in call, etc).

- **Contact list**

DID numbers can be also used to be assigned to users contact lists if they are listed as Shared DID's in the users table. In this case the DID numbers can be shared and will serve multiple purposes:

  - Incoming call from user to a DID in its own contactlist: call will be routed to the contact (username or phone number, which is "best
  - Incoming call from contact (username/phone number) to assigned DID: call will be routed to user (to which user the contact belongs)
  - For outgoing calls the A number will be also set accordingly or after the rules discussed above

- **Shared DID**

For shared DID you will have to create users with the DID numbers (username set to DID number, shared did option checked) and then these DID numbers can be used by multiple users in the same time. These will ease the access to users favorites from other networks (calling via third-party servers or via the IVR).
Endusers can assign their contacts (internal user or external phone number) to these DID numbers from webportal phonebook page.
To enable/disable this feature, you might explicitly set the feature_shareddidnumbers option to 1 (or set to 0 to disable). The default value is -1 which means that will auto turn to 1 of there are shared did numbers in the system (detected at startup, reload and nightly)

If the softswitch is used by multiple companies then you can set the BelongsToCompany field to separate the DID numbers by companies.

For incoming call to a shared DID the call will be routed to the "best" user or outbound number (Best effort. The user who have called last time the current caller). You can also use share DID's to assign a local number to called users or outbound numbers ("Quick Call" feature).
For outgoing calls a DID number will be also assigned as the A number.

Data:

Users contactlist are stored in tb_speedial:

- userid: link to "main user"
- num (any number assigned to this contact -Phone1)
- uphone2 (any number assigned to this contact -Phone2)
- uname (any user name assigned to this contact -Username)
- udid (the shared DID)

Usage:

- Add a few DID number from the "Users and devices" form. These are like normal endusers but their username is a DID number and the "Shared DID" option is checked on the "Functions" tab (this way the "ispublic" field will be set to 4 for these numbers/users)
- Set the "feature_shareddidnumbers" configuration option to 1 (the default is -1 which means that will be automatically enabled once you have added a few DID numbers, otherwise automatically disabled). You can toggle this from the "users and devices" form "functions" tab "Shared DID" option.
- Once this is done, users can assign DID numbers for their Phonebook entries from the web enduser interface. (stored in the "udid" field in the "tb_speeddial" table). This is optional since the numbers can be found also based on the CDR records (last called numbers)
- Once a DID number is dialed, the following lookup procedure takes place:

Shared DID lookup procedure:

- The calls can come from the internal network (a SIP endpoint) or from external (from PSTN via a traffic sender). The server will identify the caller via SIP authentication, A number authentication or PIN authentication via IVR. If the feature_shareddidnumbers is set then the server will do the following lookups (in this order):
- incoming call from any enduser to a shared DID in its own contactlist -> call will be routed to the contact phone or name (main user -> contact)
- incoming call from any contact (did, phone or name) to the assigned DID -> call will be routed to the main user (contact-> main user)
- incoming call from any to a shared DID which is found in last 2 hours cdr's -> route to user who was called this did last time if any (any -> main user). So if a shared number is used for a call, the called person can call back to CallerId and will reach the best suitable contact.
- incoming call from any contact (did, phone or name) -> call will be routed to the main user (contact-> main user)
- incoming call from any to a shared DID which is found in last day cdr's -> route to user who was called this did last time if any (any -> main user)

These lookups are flexible, so the caller/called/did might be found both with and without IEC or country prefix (+/00/country prefix).

- **Lookup called user on fix DID**

In some circumstances you will receive all calls from devices with the same or invalid destination number. For example you might have a VoIP-GSM gateway forwarding all calls with the target set to its phone number assigned to the SIM card or configured by the administrator.
In these circumstances the target user might be determined by a best match using the A (caller) number as previously described Shared DID procedure (looking at users contact list or last called numbers).
You can enable this from the "Users and devices" form "functions" tab "B number lookup" option (This will set the anumberlookup field to 10 or 11).

You might also set the replaceanumwithdid after your need: -1=auto (default),0=no,1=shareddid,2=companydid,3=cdr,4=all,5=4+extended cdr search

- **tb_telnumbers**

Phone numbers can be stored in tb_telnumbers and managed from the “Phone numbers” form.
This table can be used to store the phone numbers but also for special DID number storage. Editable from “Phone numbers” form.
Fields:

- id: auto increment db id
- datum: record creation date (number added date)
- telnumber: the phone number
- ttype: 0=def,1=normal,2=did,3=smsdid,other=other type
- free: 0=allocated,1=free (not allocated),2=locked,3=temporary locked
- useddate: date when it was assigned to user
- comment: any notes
- location: not used
- lockeddt: number preallocated at sign-up with the getphonenumber API
- gatewayid: set to SIP server ID if number has to be used only with this SIP server
- userid: number was last used for this user (trying to use the same number with the same user)

You can assign a DID to a user in the following ways:
-use it as the “username” field
-set the “telnumber” field
-add it to tb_users_othernumbers table (if you wish to assign more than one did to a single user)

API
See the “Assign phone number to users” section in the API documentation for the related API’s (getphonenumber/lockphonenumber/assignphonenumber)

Terms and Settings:

-DID: Direct inward dialing (DID) or called direct dial-in (DDI) is a telecom service to share a single SIP trunk among one or more users
-ANI/CLI authentication: Automatic Number Identification/Calling Line Identification
-Toll free: is a special telephone number, in that the called party is charged the cost of the calls by the telephone carrier, instead of the calling party. This can be configured as a normal access number (enduser) and eventually with higher billing (because we will be the billed party in this case)
-local DID number: normal access numbers. Usually you will have separate DID numbers for different regions to minimize enduser costs
-callback: DID or toll free number configured as enduser with iscallback set to the required IVR
-ANI callback: same as callback with User-ID based authorisation (A number)
-Virtual Numbers (DID): "real" phone numbers allocated for users. You have to buy DID numbers from CLEC or any other service provider like didx.net

TelNumber: sip telnumber.users can be contacted if we call there username or telnumber
Username: the most important field. Used for authentification and also as a DID number. This field is unique and cannot be empty.
ShortTelnumber: sip short telnumber (for example if several users has the same BelongsToCompany field)
DisplayName: how the user will be displayed. Can be null
More than one DID or ANI number can be assigned for one user. For this the tb_users_othernumbers table is used where the type field is 0 for DID numbers and 1 for ANI numbers.

autogeneratenewuseraccounts; //0=no,1=auto generate random,3=auto select from tb_telnumbers,6=select from tb_phonenumber
autogeneratedid; //(new) 0=no,1=auto generate random,3=auto select from tb_telnumbers,6=select from tb_phonenumberx
allowenduserdidedit; //allow enduser to edit its own telnumber field 0=no,1=yes for resellers,2=yes for endusers

feature_shareddidnumbers: -1=auto (def), 0=no,1=yes old,2=yes new,3=yes new and old, 4=also check last calls, 5=extended search, 6=extreme
shareddid_checkcontactlist: //-1=auto (def) ,0=no,1=yes old,2=yes new,3=yes new and old
shareddid_checkcdr: -1 (def) =auto,0=no,1=yes,2=yes extra
checklocaluserforshareddid = 1; //0=no,1=yes (def)

replacecalleroncalls: 0=no,1=default checks (best num; default),2=dbusername,3=dbsipnumber,4=bestnumber,5=first good num, 6=check minassertedidentitylen,7=with phonenumber,8=anonymous
replaceanumwithdid: -1=auto (def) ,0=no,1=shareddid,2=companydid,3=cdr,4=all,5=4+extended cdr search

forcedcallernidentity:-1 (auto, def) , user and global setting 0=no,1=default checks (best num; default),2=dbusername,3=dbsipnumber,4=bestnumber,5=first good num,6=check minassertedidentitylen,7=with phonenumber,8=anonymous

identityrewrite/identityforward: user setting to replace client->callernumber (controlled by config.identityrwmode)
identityrwmode: 0=no rewrite, 1=basic, 2=conform sip specification (identity; def)
sendidentity: 0=never,1=no (def),2=sometimes,3=always
configallowusernumbertopstn: -1=auto, 0=no,1=yes if phone number, 2=always (send call to outbound if user is offline)

cli:

user setting
0: forward always   (forward asserted as normal number always!)
1: normal handling (forward asserted as normal number) -default
2: forward as asserted identity always (forward identityrewrite asserted)
3: forward as asserted identity only to trusted domains (forward identityrewrite asserted)
4: normal hide (no idenityrewrite forwarding)
5: force hide (no asserted identity too!)

minassertedidentitylen:,7=with phonenumber,8=anonymous affects only P-Asserted-Identity; will owerwrite any cli settings. default is -1
didhandleruser: a globally configured number where incoming calls to DID will be routed if no user match (can be a receptionist or an IVR)
globaldidout: a globally configured number to set the A number to this for all calls (should be used only by small business owners)

Incoming calls can be checked against the tb_telnumbers table with the v_check_localnum to check if the target is a local number (even if it is not in the tb_users table).

4.4. Routing

A quick tutorial can be found here.

4.4.1. Dial patterns

The "dial patterns" terminology is not used with the mizu voip server. You can apply the number rewrite rules by the following ways (starting from simple to more complex rewrite rules)

1. Number normalization: Open the "Configurations" form and search for the "normalize" string to list all related settings.
2. These settings can be used for basic number "normalization" to remove strange characters (like new line) and to remove IEC code. By default this is set to remove the common international escape codes (00,+). To turn this off, change the followings:
   a) set cfg_normalizenumbers below 3 (2 should be good)
   b) set normalize_localpx to 0
   c) set "normalize_clean" value below 4 (3 should be fine)
3. Tech prefix: tech prefix related operation should be done by using the "techprefix" box in the "Users and devices" form. Set this for the "SIP servers" where a specified tech prefix needs to be sent or set it for the "Traffic sender" when you receive the traffic with a techprefix from some routes.
4. Prefix rules: this is used for simple prefix add/remove operations. (discussed below)
5. Dial plans: you should use this option if you need full control

4.4.2. Firewall

The firewall rules are checked also at transport level so this is the most effective way to block some unwanted traffic sender.
With the firewall you can block source IP addresses and caller user names (or CLI's).

For IP address rules, just enter the source IP address as the source.
For caller user rules, prefix the name/number with U:. For example to block user 'hacker', enter 'U:hacker' as source.

All source are allowed except those are listed if the ip with ‘*’ is 1.
Otherwise (if the source ‘*’ is set to 0) all address are blocked except those that are listed here.

Note:
*Make sure to do not delete the * - 1 rule.*
*If the * -1 rule exists, then it acts as a usual firewall, blocking the sources with addrenabled set to 0.*
*If the * -1 rule doesn’t exists, then it will allow only sources with addrenabled set to 1 and all others will be blocked!*

Be avare of IP address spoofing. Some attackers are capable to falsify the source IP, thus your security should not depend only on source IP filtering (additionally you should also use SIP digest authentication or at least a tech prefix even for trusted traffic senders)

4.4.3. Normalize numbers

By default the server will "normalize" all numbers. This means that it will clean the numbers from garbage characters, sql injection attacks and might remove/add IEC/CC/NEC prefix based on the circumstances and global configuration (international escape code, country code).

If left with these default settings, you will have to deal with this normalized number format for all the call processing including routing and billing.

This behavior can be quickly changed by the normalizedef setting for which you can assign the following values:

-1: default, recommended (mostly like 2) [validateinput=3; normalizenumbers=2; normalizecallednumbersnec=-1;]
0: don't touch (dangerous) [validateinput=0; normalizenumbers=0; normalizecallednumbersnec=0;]
1: minimal (clean the numbers) [validateinput=1; normalizenumbers=1; normalizecallednumbersnec=1;]
2: basic (clean the numbers) [validateinput=2; normalizenumbers=1; normalizecallednumbersnec=1;]
3: normal (remove IEC like + and 00) [validateinput=3; normalizenumbers=2; normalizecallednumbersnec=1;]
4: strict (might add CC and rewrite NEC) [validateinput=4; normalizenumbers=3; normalizecallednumbersnec=2;]
5: extra (full input checking and number rewrite) [validateinput=5; normalizenumbers=4; normalizecallednumbersnec=3;]

It is recommended to leave this setting with its default -1 value and you should change it only to have full control on number rewrite rules (set to 1) or if you wish to set a strict checking (set to 3)

Usually you should leave this (and the other related global config options below) with the default values and you might add your own special number rewrite rules by the "Rewrite" module if you need so.
If you want more control for the rules, then you should leave it with the default setting of -1 and change the global config options below.
If you want full control for the rules, then you should set it to 2 (or 1) and use the "Rewrite" module to add your rules.

Other global configuration options related to number "normalization" and rewrite are the followings:

outgoingprefixremove:
List of prefixes to be removed from the called number for outgoing calls separated by comma
For example +,00

outgoingprefixremove_maxlength:
Will apply the “outgoingprefixremove” rule only if the number length is at least this number inclusive

outgoingprefixremove_minlength:
Will apply the “outgoingprefixremove” rule only if the number length is less than this number exclusive

outgoingprefixremove2:
Single prefix to remove but within the below rules.

outgoingprefixadd:
Set any string prefix which will be applied for called numbers for all outgoing calls.
For example +.
If you need to change the prefix only to one direction, then use the "Rules".

outgoingprefixadd_minlength:
Will apply the “outgoingprefixadd” and “outgoingprefixremove2” rules only if the number length is at least this number inclusive

outgoingprefixadd_maxlength:
Will apply the “outgoingprefixadd” and “outgoingprefixremove2” rules only if the number length is less than this number exclusive

outgoingprefixadd_ifbegin:
Will apply the “outgoingprefixadd” and “outgoingprefixremove2” rules only if the number begins with this

outgoingprefixadd_ifnotbegin:
Will apply the “outgoingprefixadd” and “outgoingprefixremove2” rules only if the number NOT begins with this

All the above can be applied also when call arrives by setting the incomingprefix settings.

normalizenumbers: how to normalize
0=no,1=basic,2=normal,3=more (hu/ro),4=extra
default is 2

normalizecallednumbersnec: NEC normalization (National Destination Code)
-1=depends on normalizenumbers and normalize_localpx, 0=no,1=yes,2=extra
default is -1

loadcallednumberfromto: 0=autoguess,1=no (from first line),2=yes (from to)

rewriteinternalcalled: rewrite the called numbers for calls to endusers
0=don't set,1=don't rewrite,2=client to as received,3=client to username,4=client to telnumber,5=client to best if match,6=all to as received,7=all to username,8=all to telnumber,9=all to best if match
default is: 7 for servers and 6 for gateways

forwardcalled: what to use for request URI and To fields: 0: no (old server behaviour; callednumber decided at routing), 1: forward as received, 2: forward callednumber, 3: forward callednumber_header, 4: forward callednumber_to
default is: 0 for servers and 1 for gateways

forwardcalledtofs: same as the above forwardcalled but used only if the called party is on WebRTC

validateinput: how to clean the numbers
0=no,1=basic without sql check,2=basic with sql, 3=normal,4=more,5=extreme
default is 3

cfg_minnormalizelength: minimum length of the number to consider for extra normalization.
default is -1 which means that it depends on normalizenumbers (if normal then set to 5)

checknumnumbers: reject call if called is not a phone valid number
0=allow any calls; 1=allow only phonenumbers from H323; 2=allow only phone numbers for all
default is -1 which means that it depends on normalizenumbers (set to 1 if normalizenumbers is normal)

checknumport: check number portability rules
-1=autoguess,0=not check,1=check for changed prefix,2=check for changed number,3=check for changed domain,4=check for changed server id,9=check all
default is -1

The CC/NEC rewrite depends on the followings:
caller CC/NEC
country: country (2 chars)
iec1,iec2,iec3: international escape codes
countryprefix: default country prefix (country code)
prefix1,prefix2,prefix3: national destination codes
cfg_mobileprefixes: list of common mobile prefixes (useful for callenters to differentiate landline and mobile calls)

Deprecated settings (don't use these anymore):
normalize_localpx: replaced by normalizecallednumbersnec
setcalledtosipusername: replaced by rewriteinternalcalled
normalize_clean: replaced by validateinput
normalize_checksp: replaced by normalizecallednumbersnec

Example:
The following settings will normalize incoming numbers (from both national and international format) to international number format then will add 00 when forwarding to an outbound route (to a VoIP carrier):

Settings:
normalizedef: 3
normalizenumbers: 3
incomingprefixremove: 00,+
incomingprefixadd: 4
incomingprefixadd_ifbegin: 07
outgoingprefixadd: 00
outgoingprefixadd_ifnotbegin: 00

Example input dialed number:

+40721234567 OR 0040721234567 OR 40721234567 OR 0721234567 OR 0721234567
IEC: 00 or +
Country code: 40 (Romania)
NEC: 72 (Vodafone)

Normalized number:

40721234567

Outgoing number:

0040721234567

4.4.4. Caller ID Settings

On Caller ID we mean the number or name displayed for the called party which is sent by one of the followings sip headers:

-From header (URI and displayname)
-Contact header (URI and displayname)
-Authentication username
-Identity headers: p-asserted-identity, p-preferred-identity, remote-party-id
-Privacy settings: "privacy" header, "anonymous"

The caller-ID display is specified in the SIP and other RFC's but at the end it is up to the called party device to what caller-id to display (it can be loaded even from a local store instead of from the SIP messages sent by the caller party)

The Caller ID can be influenced by the following factors:
-the SIP message received by the client as described above
-global number normalization settings (see the "number normalization" section)
-global caller id related settings (discussed below)
-caller and especially the called user settings such as the techprefix
-Prefix Rules settings (this is already deprecated. Use "Rules" whenever possible)
-Dial plan SQL (this is already deprecated. Use "Rules" whenever possible)
-Rules (see the "Rules" section)

Settings:

The easiest way to influence the caller-id is the replacecalleroncalls setting (global setting and/or per user CallerID setting):

replacecalleroncalls:

(user and global setting)

0 No (Keep from the client)
1 Default checks
2 Use the Username
3 Use the TelNumber
4 Use the best number
5 Use the first good number
6 Replace with the Username if not a phone number
7 Replace with the TelNumber if not a phone number
8 Anonymous
9 From header
10 Contact header
11 Remote-Party-ID header
12 Best from client ep
13 Best from all (default)
14: Called party Username
15: Called party TelNumber

Default is: 13.

If the replacecalleroncalls/CallerID setting is set for both the caller and the called user, then it will prioritize the SIP Server setting if any, otherwise the caller user setting. This prioritization can be changed with the replacecalleroncallspriority setting.

replacecalleroncallspriority:

-1: auto (defaults to 8), 0: never (disable), 1: global config only, 2: called only (don't load caller setting), 3: caller only (don't load called setting), 4: prioritize caller, 5: prioritize called, 6: prioritize enduser, 7: prioritize traffic sender, 8: prioritize sip server (def)

forcedcallernidentity:

user and global setting

0=no,1=default checks (best num; default),2=dbusername,3=dbsipnumber,4=bestnumber,5=first good num,6=check minassertedidentitylen,7=with phonenumber,8=anonymous

affects only P-Asserted-Identity
will owerwrite any cli settings
default is -1

realcallernumber
just a routing variable
client->callernumber = realcallernumber;
client->origcallernumber = realcallernumber;

forcedcallernumber:
just a routing variable set by replacecalleroncalls and used by the client ep later
overwrite the client from number

forcedcallernidentityumber:
just a routing variable set by forcedcallernidentity and used by the client ep later
overwrite the client identity

identityrewrite/identityforward
user setting
replace client->callernumber (controlled by config.identityrwmode)
identityrwmode = 2;   //0=no rewrite, 1=basic, 2=conform sip specification (identity)
sendidentity = 1; //0=never,1=no,2=sometimes,3=always

cli:
user setting
0: forward always   (forward asserted as normal number always!)
1: normal handling (forward asserted as normal number) -default
2: forward as asserted identity always (forward identityrewrite asserted)
3: forward as asserted identity only to trusted domains (forward identityrewrite asserted)
4: normal hide (no idenityrewrite forwarding)
5: force hide (no asserted identity too!)

forwardauth:

user and global setting

confing.cfg_forwardauthentifications = 0; //0=no (default),1=yes,2=yes with username as callername, 3=yes with phonenumber as callername, 4=yes with username as authname too, 5=yes with phonenumber as authname too, 6=replace authorization with username but leave the A number intact, 7=replace authorization with sipphonenumber

can rewrite client auth_username, auth_password, callernumber

rewriteanumber:

user setting

simply replace client->callernumber

set to anonymous to disable the caller-id

cutanumber:

user setting

simply replace client->callernumber prefix

replacecalleronforward

global config option

used only for callforward

replacecalleronforward = 9; //0=no (default),1=yes,2=yes with username as callername, 3=yes with phonenumber as callername, 4=yes with username as authname too, 5=yes with phonenumber as authname too   9=auto

parseidentity

global config option

0=no at all, 1=normal,2=rewrite caller number,3=rewrite callername also

default is 1

parsesipdisplayname

global config option

0=no,1=callername,2=displayname,3=can be used for auth

default is 1

userealphonenumbercallerid: 0=no (default), 1=yes: will overwrite cfg_replacecalleroncalls! (will try to find a phone number to use)
useorigcallerforeu: 0=no (default), 1=yes: will overwrite cfg_replacecalleroncalls! (if call is routed to enduser, then will use normal caller)

replacecalleroncallsusercheck: default is 1
replaceanumwithdid: -1=auto (default) ,0=no,1=shareddid,2=companydid,3=cdr,4=all,5=4+extended cdr search

Also see the DID numbers section and the How to rewrite caller/called numbers FAQ point for other possibilities.
See the How to set the Identity header for more details about the Identity SIP headers.

4.4.5. Rules

Using the “Rules” for you can handle and rewrite almost all SIP fields based on any condition, especially the caller and the called number.
This form can be used to create both simple and very complex rules. Hold the mouse over the controls to get a context sensitive help.
Usage:

1. Click on the “Add” button to add a new rule
2. Name the new rule as you wish
3. Run as: it is usually fine if you leave the “Auto guess”, however you can specify it more exactly in certain conditions
4. Specify the conditions when this rule is executed (for example you might have to reformat the called number if the call is sent via a specific carrier; in this case use the CalledID to select the carrier)
5. Data: specify which data do you wish to load and work with it (for example the callednumber)
6. Check: specify additional conditions when this rule is executed (now you can examine the “data” loaded in the previous step)
7. Action: specify what you wish to do with the data
8. Set: specify how to wish to save back the result (usually “same as loaded data” since most of the case you will just wish to modify a field such as the called number)

You can also use SQL statements in action strings curly braces. See the How to set a random caller-ID FAQ point as an example.
Keywords can be also embedded in square braces. See the endpoint keywords in the Keywords chapter for the possibilities.

Additionally you can modify some global configuration options to modify the caller/called number. These are described below:

4.4.6. Prefix rules

You should instead use the “Rules” form mentioned above.
You can rewrite prefixes before they arrive to the routing by entering your preferences here.
The Mizu routing engine will accept only 3 digit length techprefixes or no thechprefix, so you must convert them here if your traffic sender will send the traffic with techprefix that are not three digit length.

Example 1:

To set up a rule which defines that every incoming number from ip 111.111.111.111 on H323 if begins with 1234 must be rewritten to begin with 56. Number 123499999 will be rewritten to 5699999.
If the “RewriteFrom” is emtly, then the “RewriteTo” fill be inserted before the number

Example 2: To remove 011 before all dialed numbers:

Protocol: 0
Type: 1
Value: empty
Rewrite from: 011
Rewrite to: empty
Number to rewrite: Incoming Called (0)

Example 3: To replace 06 with 446 for all SIP calls coming from 192.168.1.8:

Protocol: 1
Type: 2
Value: 192.168.1.8
Rewrite from: 06
Rewrite to: 446
Number to rewrite: Incoming Called (0)

Example 4: to handle the hungarian roaming prefix: 08 + SK + BK + NSN +SN you have to set the following values:

prefixrewritestr: 08X…
prefixrewritefrom: 9
prefixrewriteto: 36

4.4.7. Dial Plans

You should instead use the “Rules” form mentioned above.
During the routing process you can modify the caller and called number with the dialplan stored procedure (v_dialplan)
v_dialplan can be called several times during the routing depending on the checkdialplan1-4 global config option.
For CLI and A number rewrite it is enough to set checkdialplan4 to true (after routing) and check/rewrite the numbers using the LIKE operator (wildcards are enabled).

Input parameters:

```
@calledat TINYINT, /*1=first check, 2=after authentication, 3=before routing out, 4=after routing out*/
@protocoll TINYINT, /*0=SIP, 1=H323, 2=GSM, 3=Other*/
@fromip varchar(22), /*caller ip address*/
@fromport SMALLINT, /*caller port*/
@callerid int,  /*caller device database id from tb_users*/
@callernumber varchar(35),  /*caller number or sip username*/
@callername varchar(35),  /*caller name or displayname*/
@calledid INT,  /*called device database id from tb_users*/
@origcallednumber varchar(35),  /*called number as received*/
@techprefix VARCHAR(10),  /*called number tech prefix*/
@normcallednumber varchar(35)  /*normalized (changed) called number*/
```

The stored procedure can be called at different routing stage:
1: before authentication
2: after auth
3: before routing
4: after routing

This means that if a number has efect on authentication then you can rewrite it on stage 1, but if you need to modify the called number only for the b-leg call then it is better to call this function only at stage 4

Some of the input values are not set at earlier stage. For example when calledat is 1 then the callerid will be 0 because the caller is still not known at this stage.

Usually only the called number have to be rewritten, but you can also change the other values.
Accepted output values:

```
  -emty string: no effect
  -_REJECT: will disconnect the call
  -_REJECTPLAY,filename: play a file and disconnect the call
  -callednumber: will change the called number
  -callednumber,calleddialed,origcallednumber,techprefix,callernumber,callerid,calledid
```

Field details:
-Callednumber: will be used for further routing decisions, for billing purposes and it is stored in CDR record
-Calleddialed: will be used only for b-leg
-Origcallednumber: can be used for further routing decision (mostly not used)
-Techprefix: can be used for further routing decision. Deprecated after version 3.5
-Callernumber: can be used for further routing decision and stored in CDR record
-Callerid: you can modify the caller user if you change this parameter
-Calledid: you can modify the called user if you change this parameter

```
sql help:
  sql tutorial: http://www.w3schools.com/SQL/sql_intro.asp
  stored procedures: http://msdn.microsoft.com/en-us/library/aa174792(SQL.80).aspx
  string functions: http://msdn.microsoft.com/en-us/library/aa258891(SQL.80).aspx
  like operator: http://msdn.microsoft.com/en-us/library/aa933232(SQL.80).aspx

example:
    DECLARE @ret_callernumber varchar(35)
       DECLARE @ret_callednumber varchar(35)

       SET @ret_callernumber = @callernumber
       SET @ret_callednumber = @normcallednumber

       IF (@calledid = 456 and @ret_callednumber LIKE '1_23%')
       BEGIN
             SELECT '_REJECT'
             RETURN 1
       END

       IF (@calledid = 457 AND (LEN(@ret_callednumber) < 5 OR ISNUMERIC(@ret_callednumber) = 0))
       BEGIN
             SET @ret_callednumber = '1234567'
       END

       IF (LEN(@ret_callednumber) > 6 AND LEFT(@ret_callednumber,3) = '061')
       BEGIN
             SET @ret_callednumber = '36'+SUBSTRING(@ret_callednumber,3,35)
       END

       IF (@ret_callednumber LIKE '5222%')
       BEGIN
          SET @ret_callednumber = SUBSTRING(@ret_callednumber,5,35)
       END

       SELECT @ret_callednumber+’,’+@ret_callednumber+','+@ret_callednumber+',,'+@ret_callernumber
       RETURN 1
```

If the stored procedure doesn’t run successfully, then it will have no effect.

4.4.8. Blacklisting

List the blacklisted target numbers on the selected time interval and direction.
This query will generate high server load. Use it only in off-peak time if possible

4.4.9. Access Lists

You can define the “Blacklist” and the “Whitelist” here for the called numbers. The listing will be used in the routing depending of the actual packet “filtering” setting. Check section 4.5.1 for details regarding filtering.

Note: these filtering applies to targets and not for the source. If you wish to filter the call source, then use the caller banning module as described here.

Global setting to enable/disable black list: **numfiltering**:
-1=autoguess (default. Will be auto set to 0 if there are no blacklist entries or to 3 if there are)
0=nothing (disable)
1 =only from h323
2=only from sip
3=all (enable)

Blacklist method: **blacklisttype**:
-1=autoguess (default. Will be auto set to 0 if there are no blacklist entries)
0=nothing (disable)
1=phone number as string or prefix using tb_blacklist (slower)
2=phone number as number using tb_blacklist_int (very fast even for millions of records)
3=both

Global setting to enable/disable white list: **checkknownnumbers**:
0=no (disable)
1 = autoguess (default. Will be auto set to 0 if there are no blacklist entries or to 3 if there are)
2=yes (enable)

Database tables:
White list: tb_knowngoodnumbers
Black list: tb_blacklist

Blacklist fields:
-telnumber: country code + operator + telnum
-sure: levels

tb_blacklist.sure:
0 –probably good numbers (reput)
1 - not sure (holes)
2 - probably wrong number (monthly autodisabled)
3 - very sure (roaming numbers)
6 – always block (not only to gsm)

tb_packets.filtering: *determines how we check the blacklist and the known numbers*
0 -no filtering
1 - filter if very sure blacklisted (tb_blacklist.sure >=3)
2 - filter if probably blacklisted (tb_blacklist.sure > =2)
3 –filter if suspicious (tb_blacklist.sure > =1)
4 - filter if present in blacklist (any tb_blacklist.sure)
5 - filter if not a known number
6 - filter out if no sure known number

Blacklist can be also built automatically as defined by the builddynamicblacklist global config option:
-1: auto (will auto turn to 40 if numfiltering is enabled)
0: disable
Other: minimum failed calls per day

Failed calls are checked periodically with the following minimum count to block:
-hourly: builddynamicblacklist/2
-daily: builddynamicblacklist
-weekly: builddynamicblacklist*2
-monthly: builddynamicblacklist*5

Other global config options:
blacklist_maxbcount: max count of numbers to block one time (default is dynamic based on average number of calls per day)
blacklist_minasr: minimum asr per number to block (default is 5)
blacklist_minacd: minimum acd per number to block (default is 10)

You can add a number to the black list form the “Access Lists” form “Add number to blacklist” button or by executing the following SQL from the “Direct Query” form:

```
insert into tb_blacklist (telnumber,reason,sure,maxtryagaincount,cantryagain,blockdays,lastblocked)
values ('TELNUMBER','COMMENT',6,0,0,99999995,getdate())
```

```
(replace TELNUMBER and COMMENT after your needs)
```

Block scanning

Some spammers might try to scan a number range by making calls to similar numbers (such as trying numbers from A to Z).
Blocking these attempts can be configured by the blockscanning global settings:

- blockscanning_treshold: -1: auto, 0: no, 1+: yes, max count of subsequent similar numbers. Default is 45, or 15 if option checked in the config wizard or 0 if not checked.
- blockscanning_mode: strict (permissive; will count only subsequent numbers or diff is less then 10), 2: general (will check if diff is less then 30 or if there is only single digit change. threshold calculation will be incremented by 3 for subsequent numbers). Default is 2 (general).

Smart blacklist

Will block a number after X failed attempts in Y time interval for Z amount of time.
The “number” here means the target/called numbers, not the source/caller number.
Settings:

- smartblacklist_enable: enable/disable smart blacklist. 0=no,1=on normal load only (will not run on high server load), 2=force always. Default is 0. Set to 1 to enable.
- smartblacklist_peruser: smartblacklist is per user or for all calls. 0=no (so it will check globally), 1=yes for endusers, 2=yes for all callers, 3=yes for all users with “smartbl” field set to 1 (otherwise not checked at all). Default is 1
- smartblacklist_callduration: call duration in seconds to be considered by smartblacklist   //0= all calls, negative: count only connected calls, other: count only calls below this duration including not connected calls. Default is 5
- smartblacklist_callduration_min: specify only if you wish to check calls above a specific duration. For example if you wish to block users with calls between 1-3 seconds then set smartblacklist_callduration_min to 1 and smartblacklist_callduration to 4.
- smartblacklist_blockafter: number of attempts after to block the number. Default is 20
- smartblacklist_checkperiod: check calls in this period. 1 means one day, 2=two days, 0.5 means 12 hours. Default is 1 (one day). Increasing this might increase CPU usage.
- smartblacklist_blockperiod: block the number for this duration where 1 means one day. 0=not used (will block until goes out of the checlperiod range), other: will add to blacklist for the specified time. Default is 0.
- smartblacklist_voice: voice playback on call reject (name of the file)

Update old database:
ALTER TABLE tb_blacklist ADD [calleruser] [int] NULL DEFAULT (NULL)
ALTER TABLE tb_blacklist ADD [expireat] [datetime] NULL DEFAULT (NULL)
ALTER TABLE tb_users ADD [smartbl] [tinyint] NULL DEFAULT (0)

SP v_check_blakclist:

```
select TOP 1 telnumber,sure from tb_blacklist WITH(NOLOCK) where
```

```
@called_norm LIKE telnumber+'%' AND blocked = 1 AND telnumber is not null AND LEN(telnumber) > 0 and (expireat is null or expireat >
getdate()) and (@callerid = calleruser or @callerid = 0 or calleruser = 0 or calleruser is null)
ORDER BY sure desc
```

4.4.10. Routing

For every time period and direction a “Routing Pattern” needs to be defined. Every Routing pattern has a list of routing directories which may be Mizu GSM Gateteways, other H323 gateways or gatekeepers, ISDN gateways or SIP proxies in priority order. Set up as much directories with the same priority order as possible so the routing engine can prioritize itself after other settings (device priority, LCR, quality, BRS)

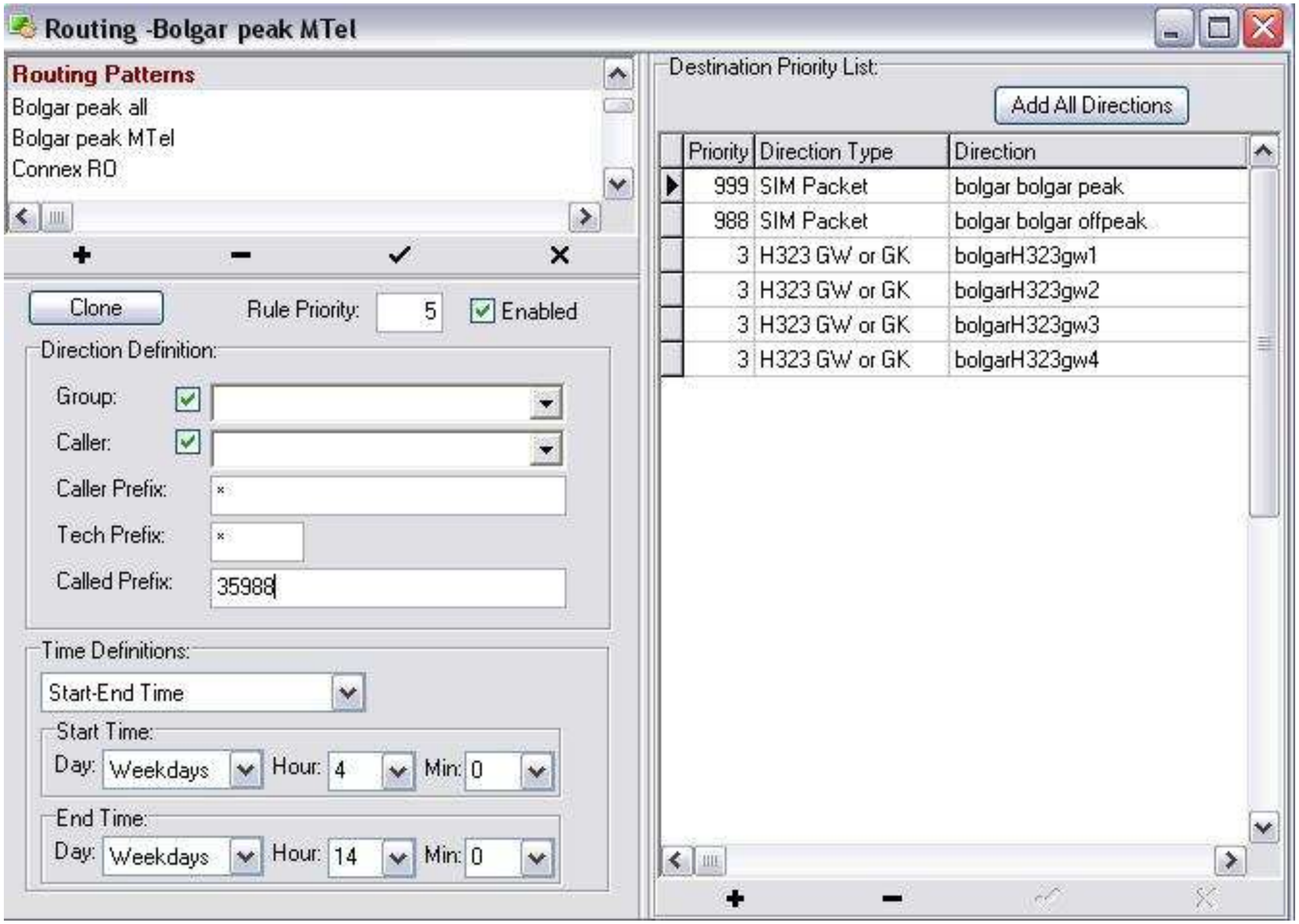
Generic rules can be defined by setting the pattern priority lower. For example for every call that doesn’t have a specific route can be routed to a specific direction (otherwise is dropped)

There is a list of typical time definition. If none of them match your needs, the “Start-End Time” entry can be selected to specify proprietary intervals.

In Caller Prefix, you can place only one prefix.

Tech prefix can be empty string, asterisk (*) or 3 digit length number.

Called prefix can be one prefix (with any length) or a list of prefixes with 3 or 4 digit separated by comma.



Routing Patterns
Bolgar peak all
Bolgar peak MTel
Connex RO

Rule Priority: 5
☑ Enabled

Direction Definition:
Group: ☑
Caller: ☑
Caller Prefix: *
Tech Prefix: *
Called Prefix: 35988

Time Definitions:
Start-End Time
Start Time:
Day: Weekdays Hour: 4 Min: 0
End Time:
Day: Weekdays Hour: 14 Min: 0

Destination Priority List:

Priority	Direction Type	Direction
999	SIM Packet	bolgar bolgar peak
988	SIM Packet	bolgar bolgar offpeak
3	H323 GW or GK	bolgarH323gw1
3	H323 GW or GK	bolgarH323gw2
3	H323 GW or GK	bolgarH323gw3
3	H323 GW or GK	bolgarH323gw4

Tip: you don’t need to enforce traffic sender rights by routing. The routing can be done as generic as possible for example by specifying only Called Prefixes (Leave the other direction option blank or ‘’). Rights can be enforced by setting “Enabled Prefixes” for all Traffic Senders

Routing Configurations

Try to set up all routing rules and prioritizations using this form.
Try to avoid prioritizations by gateways, simpackets or channels (absolutepriority, priority, allowedpartners, prioritypartners, etc)
Almost all kind of configuration can be set up by using only the “routing” form

The actual priority order of your route list (right side list) will be affected by numerous factor:
-**Failovering**: when a device or a device/direction is in failovered state, then its priority is lowered
-**Load balancing**: will prioritize the direction where the last routed date time is oldest (only if the routing entries are entered with the same priority)
-**LCR**: the route with the lowest price will be prioritized
-**BRS**: the best quality/lower price route will be prioritized

These settings can be controlled by the “brs_lcr” global config option:

0=not used
1=only lcr for not gsm
2=lcr
3=brs for not gsm
4=BRS (default)
5=BRS+LCR

4.4.11. Routing workflow

Introduction

The routing in the Mizu softswitch means deciding on which active gateway, user or gsm channel should we route the incoming call from traffic senders and endusers.
The routing is influenced mainly by the following:
-device ownership and access rights (allowedpartner settings)
-routing time, direction and the selected pattern (device/packet priority list)
-various priority settings

The routing is blocked if the following conditions are met:

General conditions
syntax error in incoming number (or not known)
max call/day, max speachlength reached (licensing option)

Caller user check
the traffic sender reached their maximum channels
caller gateway, simcard or simpacket is not enabled or temporarily disabled
failed authorization (wrong originating ip, bad username or password or wrong techprefix)
Caller “CanDial” setting is set to false
Caller tb_users.enabledprefixes not match (‘*’ allow all numbers)
Check if other traffic sender has the same ip/techprefix (caller mismatch)

Routing
direction and time don’t match a routing pattern
no active device or simpacket from the selected pattern priority list

Called device/gateway checks
Called gateway(s) is not enabled, not active, temporaydisabled, allowedpartners don’t match or any other problem
Called gateway onlytestcalls not match
Called gateway enabledprefixes not match (‘*’ allow all numbers)
Called gateway blockprefixes match
Called device filtering option doesn’t allow blacklisted number level (if the incoming number is blacklisted)
gateway has testprefix but does not match

Called simpacket check (only for gsm directions)
Packet is not enabled
Caller is not listed in allowedpartners
packets.waitaftercall second not elapsed since last call
packets.filtering. blacklist/whitelist restriction (filtering option doesn’t allow blacklisted number level (if the incoming number is blacklisted))

Simcard check (only for gsm directions)
Called simcard(s) is not enabled, temporaydisabled, not ready, allowedpartners don’t match or any other problem
partner is not allowed on the simcard (allowedpartners)
the simcard is prepaid, but it doesn’t have enough credit
two subsequent calls cannot be routed to the same simcards (configurable)
there was a credit request or recharge in the simcard in the last minute
cannot request credit from prepaid card more than 5 times
maxmonthlyminutes,maxdailyminutes,maxallminutes, maxmonthlyminutespeak are reached
no report from the channel for more than 5 minutes (the gateway may have lost its network connection or power)

Routing priority order
If emergency number, than the defined emergency route has the biggest priority
Routing pattern priority (if two or more pattern overlap)
Routing pattern direction/time best match (if two or more pattern overlap)
Called gateway Globalabsolutepriority
Called gateway and simcard absolutepriority
Positive routing priority (deprioritze simpackets with negative routing priority -these are “emergency packets”)
SimPacket absolute priority partner (absprioritypartner -if set and if match the caller)

Simcard caller priority (absprioritypartner **-**if set and if match the caller)
Gateway absolute priority
Gateway called priority (if set and if match the caller)
Simcard absolute priority
Routing list priority/100 (differences more than 100 in priority list)
Called gateway is not failovered -value lower or higher than the current date-time (for automatic failovering)
Called gateway is not failovered for the currenct called prefix (direction)
Simcard is not failovered - value lower or higher than the current date-time (for automatic failovering)
SimPacket is not failovered
Tpercek priority (hungarian tmobile specific)
Routing list priority
Elapsed time from last call disconnect is more than 10 sec
Gateway callerpriority match the caller number
Gateway prioritypartner match
Simcard priority partner match
SimPacket nopriority partner not match
SimPacket priority partner match
Gateway priority (simple)
Simcard priority (simple)
Simcard minimum monthly speechlength not reached
Simcard minimum daily speechlength not reached
Simcard desired monthly speechlength not reached
Simcard desired daily speechlength not reached
Simcard todayspeachlength desc order (simcards with more callduration has lower priority)
Simcard thismonthcallcount desc order (simcards with more callcount has lower priority)
Simcard thismonthspeachlength desc order (simcards with more callduration has lower priority)
Simcard a.creditrequestfails desc order (simcards with more failed credit requests with lower priority)
Gateway ready channels (balance calls across gateways)
Last call begin on the simcard (balance calls across simcards -simcards with the most recent calls has lower priority)
Simcard currspeachlen desc
Simcard GSM Fieldstrength
Simcard lastrectime (for randomizations)

Technical description:

Call arrive from traffic sender or enduser via SIP or H323
Check if MAXCCALS restriction reached (licensing option). Drop if yes.
Check if maxslperdayreached reached. Drop if yes.

Check if maxroutereqpermin reached. Drop if yes.
Check if the current call is a routing retry (forked calls). Drop if too much retry
Normalize caller ip addres
Check if the call was from the local SIP2H323 module. Return with the already prepared target if yes
Chech if the call was arrived from GSM gateway. (callin option). Replace caller and called after the config.
Correct the called number string if it is corrupt.
Check min/max length of the called number
Check if the incoming call is a testcall. Set the testcall flag is yes
Check and apply prefix roules (tb_pxrules – rewriting the called number)
Authenticate the caller (after username/password, ip pr techprefix). Drop the call with "no such user" reason on fail
Add techprefix if needed
Setup sip parameters if the call was arrived from sip
Normalize called number (check prefix, area code, etc). Drop if wrong number
Check if subsecvent wrong call
Check if the caller exceed its max line restriction
Check blacklist and whitelist
Check the embedded firewall
Check if caller called itself
check if called if a sipuser (Username, telnumber, short telnumber)
check the forwardalways option
check the ringgroup option
setup called endpoint if found
get time variables (peak, holiday, etc)
Get the correct routing pattern
Check routing list in priority order
If simpacket found, than Check simrouting
Drop the call if no route found

4.4.12. RADIUS

Define the radius servers, protocol and login information here. Used for authorization and billing.

4.4.13. BRS

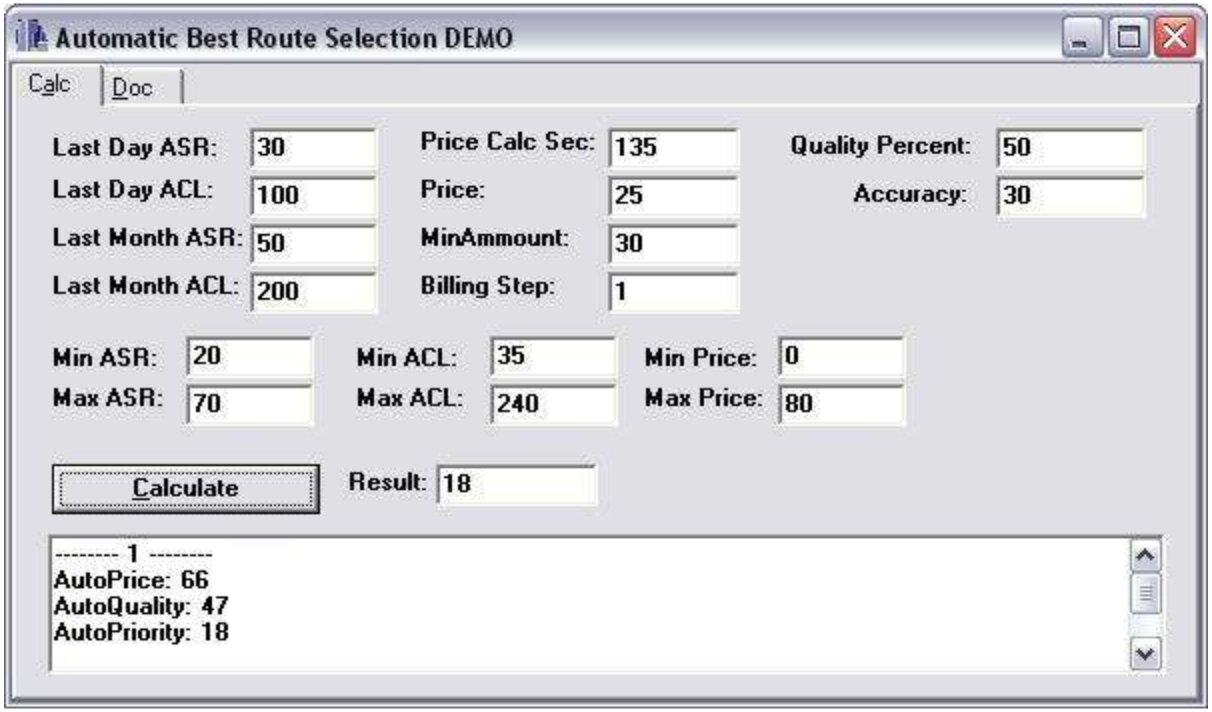
Short name for "Best Route Selection".
In addition to LCR (Least Cost Routing), the Mizu routing engine can take in consideration the quality of the route.

To activate BRS based routing, the "brs_lcr" global configuration option have to be changed to 4 or 5. Then the routing will check both the pricing and the routing, which you can fine-tune using the "BRS" form (only after you have some traffic, so the table is populated with the statics. If you would like to change the default values, you can do this only directly in the database).
Finetuning means the change of the following fields: accuracy,minprice,maxprice,minasr,maxasr,minacl,maxacl.
The rest is populated automatically based on the call statistics.

If you would like only LCR, simply set the “Quality Percent” field to 0.

If we put some directions with equal priority in the pattern, then the system will choose the routing automatically depending on price and quality, when other settings don't modify the routing (min/max minutes, gw/sim priorities, failovered directions, etc)

The server automatically calculates am „autopriority” on every route. This priority is the combination of the quality and the price. The quality is calculated as an average of daily asr/acl and monthly asr/acl. The price is calculated to pricecalcsec seconds with the given minammount and billingstep (from packet prices). The server route the traffic on the higher priority direction, BUT it will try the other routes periodically (to check if quality have changed). You can change this „next time try” setting by changing the values of TryedCount,NextTry, NextTryCount. If the best quality gateway for a route will change, then we will reset TryedCount,NextTry and NextTryCount values to their defaults. (so the server can recheck quicker)



Last Day ASR:	30	Price Calc Sec:	135	Quality Percent:	50
Last Day ACL:	100	Price:	25	Accuracy:	30
Last Month ASR:	50	MinAmmount:	30		
Last Month ACL:	200	Billing Step:	1		
Min ASR:	20	Min ACL:	35	Min Price:	0
Max ASR:	70	Max ACL:	240	Max Price:	80

Result: 18

```
-------- 1 --------
AutoPrice: 66
AutoQuality: 47
AutoPriority: 18
```

Fields have the following meanings:
-Id: database identifier. Auto increment
-Gateway: gateway id (called)

-Direction: called prefix
-QualityPercent: how much the quality will contribute to the final result. If price is very important for us, set this value lower. Default is 50%
-Accuracy: how accurate the final result will be. If we set it too high, then we probably will have only one route as the best. If we set it too low then too little discrimination will be made between routes. So, the final result (AutoPriority) will be lowered only if we have too wrong acl, asr or price.
Default is 30%. This default means that the AutoPriority will change only if price will change with 2-3 amount or asr will change with at least 15% (considering asr between 10 and 80, price between 0 and 40 and QualityPercent as 50%)
-ASRDay: last day ASR (automatically calculated every day)
-ACLDay: last day ACL (automatically calculated every day)
-ASRMonth: last month ASR (automatically calculated every month in the last day)
-ACLMonth: last month ACL (automatically calculated every month in the last day)
-MinASR, MaxASR, MinACL,MaxACL: when asr or acl reach the min value, then the line is considered very wrong. When it reaches the max values, then the line is considered very good. Must be configured manually for every direction, because the statistics will change dramatically by country
-MinPrice, MaxPrice: min-max prices/minute. set it to a very wrong price to that direction and the max value to a very good one. Calculate it with consideration to billing step and min minutes (so you must fill in as 1/1 price)
-PriceCalcSec: we estimate the price values with this value to get a gross value
-TryedCount: how much time we have tried this alternative route until now. Helps us the decide how to increment NextTry. It will grow only until 7
-NextTry: we will route calls to this route beginning with this date. Will grow exponentially until 1 month.
-NextTryCount: we will route NextTryCount calls on this route next time. (> CurrTryCount)
-CurrTryCount: counter to know how many times we have routed in this direction
-AutoPriority: the current priority calculated from these values and from the price settings (the result)

To see how much a parameter change will modify the final AutoPriority value, you can find a demo named AutoPriorityDemo in MManage, Config menu - > Utilities. Before changing any value in the BSR table, please play a little with this demo.

4.4.14. Failovering

Mizu server and gateway will make automating failover between outbound gateways if the “CAN_failover” and “hasfailover” global configuration settings are set to true (they are true by default).
The following failovering procedures are done by the mizu voip server:
-gateway failover: when an outbound gateway has wrong global statistics
-direction failover: when only a few direction (prefix) statistics are wrong on an outbound gateway
-failover on subsequent called number
-in-call failover (this is also called as “**rerouting**”)

The failover module will collect **statistics in the background** and will detect if a gateway is below the predefined thresholds. This means that its priority will be automatically lowered by the routing module.
Failover can occur on both gateway and gateway-direction level. When the ASR, ACD, etc is “wrong” for all calls trough a gateway, than the whole gateway is failovered. But there are situation when a gateway can handle most of the traffic gracefully and will fail only to a few direction (direction means 4 digit

prefix). In this case only these directions are failovered (this means that even if the gateway is with high priority to these directions, other gateways will be favorized –if you have enabled other gateways in these directions). The failovering are based on ASR, ACD statistics and if there is predefined subsequent wrong calls to a direction. Once a route is marked as failovered, it will be probed time to time to check if the problem was solved. This is happening only a few times (with exponentially increased time intervals) than if the quality is still wrong, the route will be marked as permanently failovered. This routes can be **reset**ed only manually from the failovering form if you set the MaxSubsFail, NoPriorityCount, NoPriorityCountD to 0 and the NoPriority to a date in the past.

The rules can be defined using the “Failovering” form. The table will be populated automatically after your traffic pattern.
You can check the route status also from here.
The following fields are defined:
ID: database id. Auto increment
GatewayID: called gateway or sipproxy
Direction: called direction (prefix)
MaxSubsFail: if we get more wrong calls than MaxSubsFail we failover to the next route if any
MinASR: if we get more lower ASR than MinASR we failover to the next route if any
MinACL: if we get more lower ACL than MinACL we failover to the next route if any
MinCallCount: we calculate ASR and ACL statistics only if we have MinCallCount cdr
SubsFails: current subsequent wrong calls detected
NoPriority: We have done a failover until this date. When the time elapses, we try this route again. This will grow exponentially.
NoPriorityCount: we have failovered NoPriorityCount until now because of SubsFails. The bigger is NoPriorityCount, the longer we do the deprioritization (NoPriority)
NoPriorityCountD: : we have failovered NoPriorityCount until now because of statistics
Manual: all routes will be added automatically to failover table with a minimum of quality requirements
Enabled: failovering enabled
Datum: record insertion or last modification date
Comment: why was the record modified last time (reason)

There are some **conditions** for the gateway and direction failovering to work correctly:
-the server must have enough call to calculate relevant statistics (this can be fine-tuned from the “Failovering” form) ; the default are optimized for medium traffic amount
-some time must elapse for a route to be marked as failovered (too aggressive settings might result in false failovering)
-you must have at least one secondary gateway/direction where the traffic can be failovered

You should not build your entire business based on the correctness of the failovering module. Outbound gateways should be **monitorize**d on a regular interval (Statistics -> by Called gateways) and you should take remedy actions when the statistics will drop to any outbound gateway (fix the problem or remove it by removing from the routing or set as temporary disabled)

Another failovering type occurs when there is at least **two subsequent calls to one number** and the first call length was below 25 seconds. This is a convenient way to detect if somebody calls a number with wrong voice quality and will call it again in a short time.

Other important failovering process is the **“in-call failovering” or “rerouting”.** This means that if the call is rejected by the first route, it can be immediately routed to the next route (without the caller to be disconnected)

This can be enabled by the configuration wizard or from the global configuration by the following options.
-allowreroute: -1: auto: if multiple routes found, 0: no, 1: yes-auto on reject, 2: yes-auto also on answer timeout, 3=yes-must always
-maxreroute: how many time a call can be rerouted
-rerouteon: on which conditions a call will be rerouted (0=disabled 1=onlyifset (on busy, on noanswer) 2=on server timeout 3=on server error 4=on all errors)
-reroutedisccodes: disconnect codes to be received for the rerouting
-noreroutedisccodes: define disconnect codes when there will be no rerouting
-rerouteconnected: specify if connected calls can be also rerouted (-1: auto, 0: no, 1: half, 2: yes, 3+ max call duration)

The rerouting is usually enabled by default (depending on your server type and enabled modules) or you can enable it by modifying the above settings. All these setting are set by default to optimal values which might be modified after your requirements.

Note: failovering will occur with increased thresholds (more slowly) if the priority between the routing directions (SIP servers) is more than 100.

4.4.15. Channel reservation

Skip this chapter if you are not using VoIP-GSM gateways.
Best quality (ASR and ACD) SIM channels can be reserved for sip or h323 originated calls.
In the sim table the **reserverfor** field can have the following values:

0=cannotreserve: this channel will not be reserved
1=sip: always reserve only for SIP (manually assigned)
2=h323: always reserve only for H323 (manually assigned)
3=ISDN: always reserve only for ISDN (manually assigned)
4=dynamic: can be allocated by the server dynamically (hourly check) -this is the default value
5=sipdynamic: allocated automatically for SIP
6=h323dynamic: allocated automatically for H323
7= ISDNdynamic: allocated automatically for ISDN

For every simpacket you can restrict the maximum **allowed** reservations by the maxalloc field. *(useful to not reserve all channel from the same simpacket)*
To setup the channel reservation use the following configuration values (mserver, simplatform config):
-reserveforh323: reserve capacity for h323. reservations will be disabled if less than 1

-reserveforsip: reserve capacity for sip . reservations will be disabled if less than 1

By example if you set the reserveforsip field to 5, you can be sure that 5 channels always remain free to be used by calls received with SIP protocol (H323 originated calls didn't consume all your channels)

4.4.16. Number portability

The number portability module can be used to alter the routing if the number is ported to another operator for MNP (mobile number portability) or LNP (local number portability) reasons.
More specifically it can influence the routing in one of the following ways:

- route the call to the server specified by newdomain:newport (*The fwdtootherdomains must have to be set to at least 1 for domain routing to have effect.*)
- route the call to the SIP server specified by sipserverid (or by carrierid with a lookup from tb_carriermap)
- change the called number as specified by the newnumber field
- add a prefix to the called number as specified by the providerpx field

You can set the ported numbers form the MManage "Number Portabilty" form or you can automate the process by programmatically changing the tb_portednumbers database table. The Export/Import wizard from the file menu can be used to easily populate this table (from CSV or other sources).

You can control the server number portability handling by the "checknumport" global configuration:

-1: auto (will turn to 4 if there are entries in the tb_portednumbers table)
0=not check
1=check for changed prefix only (replace)
2=check for changed number only
3=check for changed domain only
4=check for changed server id only
5=check for changed prefix only (insert)
9=check all

tb_portednumbers:
id: autoincrement database primary key
number: original (normalized) called (B) number
sipserverid: change the SIP server id to this
carrierid: change the SIP server id to this via the tb_carriermap lookup table
providerpx: new prefix (for example instead of 3630 changed to 3620)
newnumber: the changed number
newdomain: the new service provider ip or domain
newport: service port (defaults to 5060)
priority: checked for duplicate numbers

datum: record insertion date

You must have the providerpx OR sipserverid OR carrierid OR newnumber OR newdomain:newport specified.

If certain numbers must go to certain carriers (SIP servers) then you might use the tb_carriermap table to add the carriers and just use the carrierid in the tb_portednumbers table. This way you can easily manage the carriers used for LNP.

An important step for the number portability configuration is the access of the number mapping data. This can be set in the following ways:

- Add the number mapping manually
- Create some tool/script to download the data into the mizu table database (and run in periodically from scheduled tasks)
  The data can be extracted from a remote database, http/ftp file download or via an API
- Access the data via an API at runtime (HTTP GET)

Country specific:

Country specific operator lookup and LNP lookup can be configured with the numporttype global config option.
Currently the following values are defined:

- 0: default
- 1: Brazil specific
- 2: Mexico specific

For more complex number portability handling with CADUP and SPID rewrite per operator, you can also use the tb_routingprefix and tb_directions table to store these informations (spid and cnl fields). A detailed description about the whole process can be found here.

You might use the mimport application to import the numbers (will require customization for you data source and format).

The current implementation is for Brazil, but that can be adjusted for any country with similar requirements for LNP.

For Mexico specific LNP see the number_portability_mexic.pdf guide and download the mexic specific mimport from here.

4.4.17. ENUM

Enum lookups can be used to route the calls to other SIP servers directly bypassing the PSTN network to reduce the costs and improve the quality of the calls.

4.5. Billing

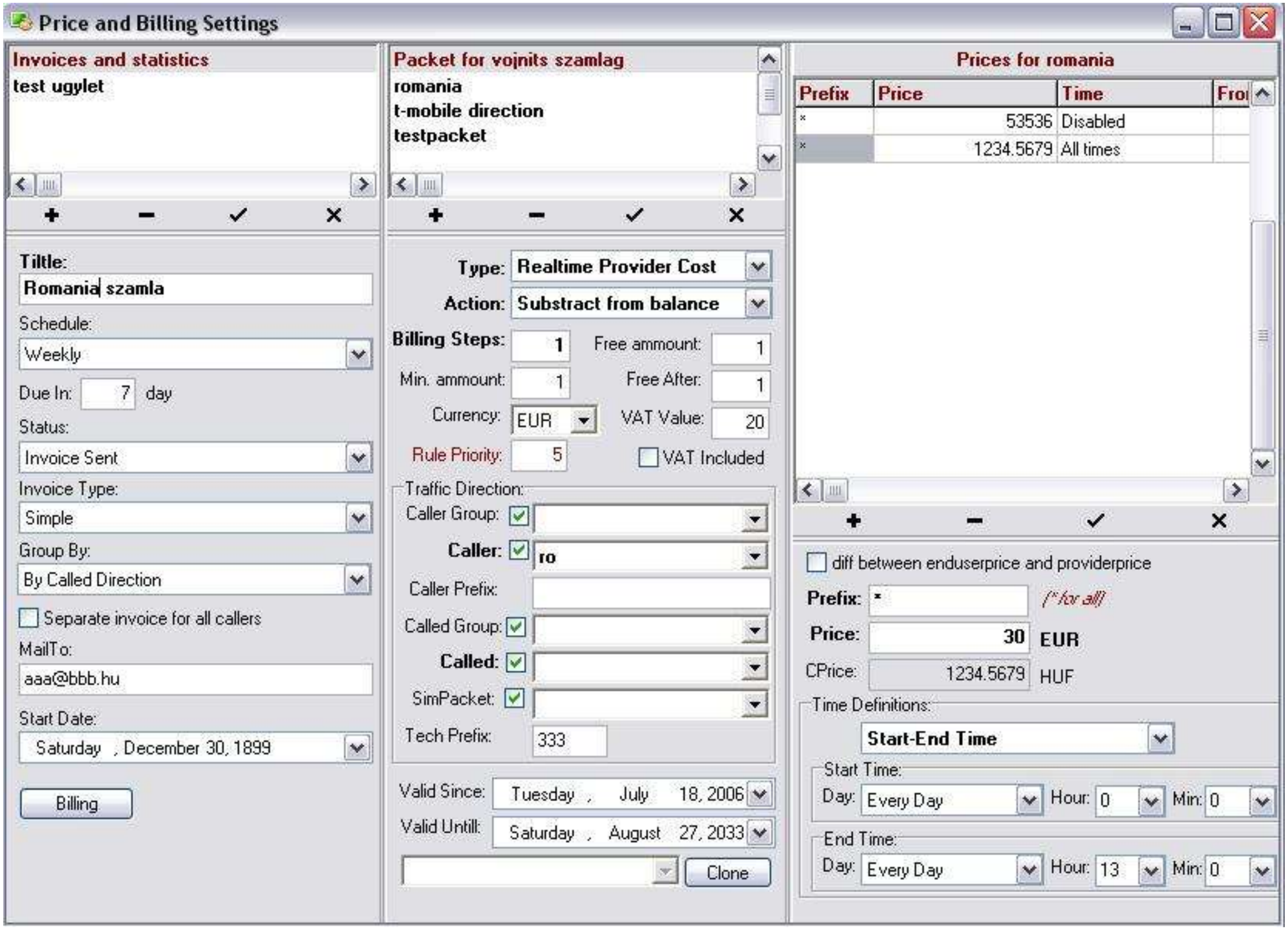
Mizu Servers and Gateways has built-in prepaid and postpaid billing.
The pricing must be set from MManage –Prices Settings form
You can list and compare the prices for different directions in the Price List.
The Billing and Invoice generation are done from the "Billing" form.
A quick tutorial can be found here.
The Billing module conforms to the Hungarian laws and can be modified to fit for other countries.

4.5.1. Price Settings

Pricing of the CDR records are done after the prices defined on this form.
You can define "**Price Groups**". All price settings that belong together after some logic (enduser prices for example). This is located in the left side of the Prices form.
~~Invoices can be generated automatically by the server and send by email, or can be loaded manually by using the "Billing" form.~~
~~You can schedule when to send the invoice or report to the partner or for you (defined by the mailto entry)~~
~~The report format can be defined by "Invoice Type" and "Group by" fields.~~

Below a "Price Group" you can have several Price Setups named "**Directions**" or "**Packets**" (the middle column in the form)
For example "Traffic from MizuTech SRL" or "Traffic to T-Mobile direction"
Here you have to set up the actual prices. The price setup is further divided into different prefixes (the right side of the form **-Pricelist**), because it is very common that you have lots of directions in a provider pricelist.

An alternative method to assign a price list to a user is by using the "Users and devices" from -> "Billing" tab -> "Billing packet" setting. This will always take precedence over the packets set in the "Price Settings" middle column. This is used to simplify the pricelist assignment for users by resellers.



Price and Billing Settings

Invoices and statistics

test ugylet

Tiltle: Romania szamla

Schedule: Weekly

Due In: 7 day

Status: Invoice Sent

Invoice Type: Simple

Group By: By Called Direction

☐ Separate invoice for all callers

MailTo: aaa@bbb.hu

Start Date: Saturday , December 30, 1899

Billing

Packet for vojnits szamlag

romania
t-mobile direction
testpacket

Type: Realtime Provider Cost

Action: Substract from balance

Billing Steps: 1 Free ammount: 1

Min. ammount: 1 Free After: 1

Currency: EUR VAT Value: 20

Rule Priority: 5 ☐ VAT Included

Traffic Direction:

Caller Group: ☑

Caller: ☑ ro

Caller Prefix:

Called Group: ☑

Called: ☑

SimPacket: ☑

Tech Prefix: 333

Valid Since: Tuesday , July 18, 2006

Valid Untill: Saturday , August 27, 2033

Clone

Prices for romania

Prefix	Price	Time	Fro
*	53536	Disabled	
*	1234.5679	All times	

☐ diff between enduserprice and providerprice

Prefix: * (*for all)

Price: 30 EUR

CPrice: 1234.5679 HUF

Time Definitions:

Start-End Time

Start Time:
Day: Every Day Hour: 0 Min: 0

End Time:
Day: Every Day Hour: 13 Min: 0

Field descriptions:
Title: the name of the invoice group

~~Schedule: how often the report will be generated~~
~~DueIn: allowed time for payment in day (used only if the report is an invoice)~~
~~Status: billing status~~
~~Invoice Type: specifies the format of the invoice~~
~~Group By: specifies the format of the invoice~~
~~Separate by caller: every caller will receive a separate invoice (used for billing to end-users)~~
~~MailTo: list of email address where to send the generated report~~
~~Last Invoice Sent: date-time when the invoice was emailed to the recipient~~
~~Last Payment Received: date-time when the payment was received~~
Direction name: name of the billing entry
Type: specify the type of the price. For example the prices used when billing to endusers, or our minute costs to service providers.
Price calculations will be saved directly in the CDRs, thus can be used in prepaid billing. In the CDR records, the following fields are used for price calculation:

-costprovider: used to calculating the minute price they needs to paid to service provider (Tmobile for example)
-costenduser: used for billing to our endusers (sip endusers, traffic senders)
-costcompany: can be used for profit calculations
-costsales: sales commission. If not set, than will be calculated by the commission value in users settings
-costother: can be used for any custom price calculation

~~Action: How to handle the calculated price in reporting. For example in "profit" calculations whe have expenses (prices paid for service providers) and incomings (from our endusers). Thus we can simply subtract the expenses from the incomings to get the "profit"~~
Billing Steps: provider specific billing interval in sec
Min. Amount: the minimum payable duration in sec
Free Amount: you may have packets when the first X second is free
Free After: you may have packets when after X seconds the conversation is free
Currency: different providers may have different currency. Used for billing.
VAT Included: if the pricelist applied for this user is with VAT included. Set to 0 if VAT is not included. Used for billing.
VAT Value: the amount of VAT applied for the pricelist. Will have effect only if "VAT Included" is checked
Convert to NET value: if you have defined the pricelist with included VAT, you should check this option, otherwise you overcomplicate the billing process. Thus the VAT value will be subtracted from the price, and you will have NET values in CDR records (try to use net values whenever possible). If you need to generate invoice, make sure that all your prices are set without VAT (net) or you have checked the "Convert to NET value" checkbox.
Convert to XXX: if you have defined the pricelist in other currency than the native (configurations->currency), than your prices will be automatically converted to native currency in CDR records.
Reseller: price created by resellers. Usually should be changed only by resellers (from web portal)
A leg grace: used for IVR billing. If the user will spend too much time using the IVR.
Traffic Direction: here you have to define the rules when the current pricing will be applied

Usually only one field needs to be specified here (for example all traffic from MizuTech SRL -caller)
The caller field will check the caller parent also, but the called field will not check the parent.

These directions will be checked in priority order on routing and billing

ValidSince, ValidUntill: the pricelist may be applied only after a specified date-time
Prefix: called number prefix (this will be loaded after "best fit"). Set to '*' to be applied to all directions
Price: the actual price
CPrice: the price converted in your currency ("currency" entry in the Configuration form and converted after the values specified in the "Currency Converter" form)
Time Definitions: the time period when this rule is applied
Diff between enduserprice and providerprice means that price will be calculated by extracting the provider cost from the enduser cost for an already existing cdr record. Cannot be used for realtime (prepaid) price calculations. Usually used when calculating "profit" values.

By clicking on the "**Clone**" button, you can easily duplicate a price list (it is very useful when you have to add only a few modification to a long pricelist)
The Billing button is a shortcut to the billing form (does not make the billing automatically)

Importing price definitions from file are done by clicking on the "**Import from file**" button.
First of all you must create a packet (middle column) defining the general packet properties and the conditions when it will be used. Then you can import a price list for the packet (right column -> "Import from file" button).
The imported file must be comma separated CSV file format.
You can use the Excel -> Save As functionality to export any Excel sheet as CSV file. You can also easily edit any CSV file in Excel. (Don't leave empty columns before the columns with data)

The file must have the following fields (columns), in this order:

1. prefix (direction definition; mandatory)
2. price (flat price; mandatory)
3. peak price (if a separate preak time price is required; you might also use a separate packet for peak/offpeak)
4. offpeak price (if a separate off-preak time price is required; you might also use a separate packet for peak/offpeak)
5. billing step (if you need different billing step by prefix instead of defining the same for the whole packet)
6. minammount (if you need different minimum billed amount instead of defining the same for the whole packet)
7. mindigits (this prefix will match only if the called number length is at least mindigits)
8. maxdigits (this prefix will match only if the called number length is at most maxdigits)

Only the first two columns (prefix and price) are mandatory. The rest can be omitted or their value can be empty.

If you use the "default peak time definition", the peak settings will be loaded from the global configuration (peaktimebegin and peaktimeend values). If this is not suitable (different service providers may calculate with different peak-offpeak definitions), you can set up the peak time definition manually (start – end hour).

If you use flat price, than leave the peak and offpeak price fields empty and vice-versa.

Importing price files may take some time, depending from your network connection speed.

Pricing example:
-a separate provider cost tariff for each of your carriers (where you are forwarding your outgoing calls)
-one enduser cost tariff for your wholesale traffic senders (or a separate tariff for each of them)
-one enduser cost tariff for your enduser or create a few groups (like premium and regular) and setup different pricing for each of them

4.5.2. Price List

On the **List** tab you can list all prices for a packet (by using the "Packet" list box) or to a direction (by entering a direction name or a prefix to the filter box)
On the **Least Cost** tab you can compare the prices from different service providers.
The Reference Packet usually is the price for your end-users.
Only peak (max) prices are compared for every direction.
On the **Directory Check** tab, lookups from the directory table are possible (directory name – prefix match).

4.5.3. Billing

The server automatically calculates the price field for every incoming CDR record, based on price settings (Section 4.5.1)
The following prices are calculated:

-enduser cost: used for invoicing for costumers
-provider cost: cost that needs to be paid for service operators
-sales cost: sales commission. If not defined in price setup, then will be loaded from users' settings ("commission") if any
-company cost: usually used for profit calculations
-other cost: for any other purpose

Billing can be done from
1. the "Users and Devices" form, Billing tab, by clicking on the "**Generate &Invoice or Report**" button (billing for the actual user)
2. set up required directions and click on the "**Billing" form** (in this manner, billing reports can be generated for more users)

The billing process will always take in consideration the selected date interval.

Billing form:
1. On the **Customized Billing** tab after selecting the required date-time interval and direction, the prices are calculated after predefined parameters (price/minute, billingstep). So you can do simple calculations using this form.
2. The **CDR Prices** tab will load the "enduser cost" and "provider cost" directly from cdr records (already calculated after real-time price settings)
3. **Generating Reports and Invoices** tab
Used for billing and reporting.

Fields explanations:
Provider: you must select the invoice emmitent here. By clicking on the "…" button, you can customize the company invoicing details.
Delete old invoices: if checked, than will clear the invoice files directory before saving the new ones.
Include inactive users: uncheck this checkbox, if you don't want to generate reports (invoices) for inactive users (inactive for the selected period)
Include child users: for example you can select a Reseller as direction source, and all "child" users will be included in billing (where the parent id will point to that reseller)
Include CDR records: include call detail records in appendix
Language: language of the invoice
Grouping: you can select any grouping options to be generated as appendix for the report
Price values: select the price field from the CDR record after which the billing are done.
Reporting: you can automatically save the generated reports or invoices to file, or open it one by one (you can decide what to do for every report -save, print or just preview)
Format: file format (text, pdf) or printing
Real Invoice: if you would like only a quick report for the selected user(s), you can do it by setting this option to "Don't generate real invoices". If you choose to generate real invoices, than it will take special processing for it (required for bookkeeping)
If you have selected a reseller, you should choose the "For Resellers" option. In this manner a real invoice only for the reseller company is generated. (A report will be generated for all child endusers, but those report are skipped from the bookkeeping)
Invoice Comment: any comment here. This will not be shown on the report
Money Precision: how many floating point digit would you like in money fields.
Completion date: defaults to the end of filling period if not modified
Method of payment: can be specified here, or loaded from user setting.

By clicking on the "&Generate report for the selected directions" button, you can generate the actual invoice(s)

4. **Invoices and Payments**

The invoice records for the selected user(s) are in this form. You can watch the debt for every user by checking the topmost record debt value.
By right clicking on a invoice record a menu will appear. From

5. You can change the price settings whenever you want, but don't forget to **Rebill** your CDR records after the new settings. All CDR prices will be recalculated for the selected time interval and direction. Users and simcards credits will NOT be modified by rebilling!

6. **Individual invoice**

On this tab you can create invoices manually (not automatically generated from cdr records)

Note: prior to generate pdf report you should configure the installed print to pdf driver to save automatically in the specified directory. The default pdf printer can be configured in the MManage menu on the Settings-> Options from. The "pdfcreator" driver is included in the MManage install package. For printing jobs, the default configured printer will be used.

The **automatic prepaid credit expirity** can be configured with the following settings:

~~Creditunit: how many credit means 1 day~~

Creditelapseunit**:** prepaid credits will be elapsed automatically after this period is elapsed. Creditelapseunit means the credit amount for 1 day. For example if you set it to 40 and the user will buy 5000 ft credit, than it will elapse after 4 month

Maxcreditelapsedays: max number of days when the credit will elapse

Accelapsedays: the number of days from creditelapsedt when the account will expire (account number will be suffixed with _elps ans set to temporarydisabled)

tb_users. Creditelapsedt: date-time when the credit will be expired

tb_users.Accelapsedt: date-time when the account will be expired

For this to work, you must set the "accountcanelapse" global config option to "true". The credit expire date is calculated when the credit is added.

**You can set prepaid credit by the "Add with elapse" button to elapse automatically.*

Monthly payments can be set for users by completing the tb_fees table. Can be accessed from the Users Form -> Billing tab.

The following fields are defined:

Userid: the user where the payment belongs

Datum: record insertion date

Name: the title of the payment

Value: net price

Usable: can be calculated in minute price (1 if yes, default is 0)

Ival:

-1: one time

0: monthly as soon as possible

Other: internal in days

Lastbilled: last time when the it was invoiced

Description: any comment here

To make the billing stricter, you might also set the following global parameters:

blocknotbilledcalls: 0=no (default),1=if best match packet is not found,2=if no exact packet match with prefix,3=block if not assigned directly for the user,4=check also tarifflist prefix,5=check also parent tarifflist

blocknotprofitablecalls 0=no (default),1=yes,2=yes also if equal,3=block also when it is not set

vgetpriceexactmatch: 0=no (default), 1=yes for resellers, 2= yes for all

notbilledcallerr=0 0=default (error with 2 priority),1=error,2=critical

4.5.4. Currency Conversion

You can set currency in 3 places:

1. -native currency in the global configuration "currency"

   This will be the default currency for all internal operations
2. -currency for pricelist packet

   When you receive pricelist in other currency, with this setting you can easily convert it to your native currency
3. -currency for the individual users

   Useful when you need to send invoices in different currency format.

The price in the cdr record will be set based on the “currencyconversion” global configuration value which has the following possibilities: 1=native currency, 2=pricelist currency, 3=user currency if match, 4=user currency

The most easy and simply way is to set the same currency in all places.
Otherwise you must refresh time to time your currency converter table.

Currency converter
Defines the conversion between your native currency and other currencies used in price settings. You should update this conversion prices as many times as possible.

Currency precizion
You can control the money precizion display by the **tb_currency_precizion** table accessible from Billing -> “Money Precizion”

Id: database autoincrement primary key
Currency: name of the actual currency (for example: HUF, USD, EUR)
Precizion: number of fractioan digits on the invoices
Final Precizion: the precizion of the “Total Payment” display
Final Rounding: rounding in the “Total Payment” (ex: 1 or 5 ft)
Separator: usually ‘.’ or ‘,’

4.5.5. Finances

You can use this form for your cash-flow administration regarding your voip business.

4.5.6. Pin codes

Recharge codes used if you have prepaid cards printed.
You can generate random prepaid codes here.

Prepaid account can be charged over the website or by ivr:
Website operation:
-user authentication (tb_users.username and password)
-check if pincard is valid (tb_prepaidcodes)
-increase credit for the user (tb_users.credit)
IVR operation:
-automatic user authentication based on sip registration or require entering the phone number to be charged
-require pincode
-check if pincard is valid (tb_prepaidcodes)
-increase credit for the user (tb_users.credit)

-goodbye message

4.5.7. The billing process

Every cdr record are handled by the billing module. Prices are determined by the v_getprice stored parameters.
v_getprice parameters:
@type tinyint: type of the billing. 1=enduser cost, 2= provider cost, 3=sales cost, 4 = company profit, 5 = other cost
@callednorm varchar(26): normalized callednumber. example: 36301111111 (B number)
@callerid int: database id of the caller user
@callernum varchar(26): caller number (A number)
@techprefix varchar(4): 3 digit tech prefix if exists
@calledid int: database id of the called user
@calledpacket int: simpacket if exists
@timetype1 tinyint: time period
@timetype2 tinyint: time period
@timetype3 tinyint: time period
@timetype4 tinyint: time period
@currday smallint: weekday number (Monday is 1)
@currhour smallint: call midtime hour
@currmin smallint: call midtime minute
@parentid int = 1: database id for the parent of the caller user

timetypes are considered when you doesn't set a concrete start-end period in the price list, and you choose from a predefined pattern (peak, offpeak, weekend, weekday, holiday,evening, night). the following timetypes are defined:
0: Disabled
1: Start-End Time
2: Peak
3: Offpeak
4: Weekday
5: Weekend
6: Offpeak and weekend
7: Evening
8: Night
9: Holiday
10: All times
11: Other Times (Rest)
example: v_getprice '1','36301234567',6555,'003615555555','150',666,-1,4,2,99,99,4,11,41,500

The v_getprice stored procedure will **return** the following fields: tb_billentries.currency,tb_billingtimes.isdiff,tb_billingtimes.origprice,tb_billingtimes.price, tb_billentries.billingstep, tb_billentries.minammount,tb_billentries.freeammount,tb_billentries.freeafter,tb_billentries.vatincluded, tb_billentries.vatvalue, tb_billentries.converttonative,tb_billentries.converttovat

According to returned billing settings, the price is calculated accordingly.

If there are no price defined, than default price are loaded. (if set)
If there are no **sales** price defined, than will be loaded from user setting (commission)
For **enduser** prices, the discounts and userreductions are applied if set so. Then the user **credit** is updated.
If there are any **error** with the billing process, than the default prices are applied if exists.

<u>Billing verification:</u>
List the required CDR records with "Show minute price option"
You can find the billing logs if you search for the called number in the server logs.
Copy the required v_getprice log in the direct query form.
Check if the returned values are as expected.

4.5.8. Invoice and payment storage

Invoices and payments are stored in **tb_invoices** in the database. *So the invoices can be searched, recreated and storno invoice can be built based on existing invoices (conforming to Hungarian laws).*
The last invoice number are loaded from database before every new invoice and incremented. Thus the invoice number increment is guaranteed the by database engine transactional behavior.

The following fields are defined:
ID: autoincrement database primary key
Type:

0=All or Recreate (technical)
1=Report
2=Proform
3=Advance
4=Invoice
5=CreditNote
6=Storno
7=Correction
8=Payment (technical: payment received)

Copynum: printed copies
CompanyID: emmitent company ID (tb_billsources)
UserID: costumer ID from tb_users (if any)
User_name: costumer name or company name

User_address: costumer address
User_regnum: costumer registration number
User_euregnum: costumer eu registration number
User_bank: consumer bankdetails (cont and address)
Payment_type: mode of the payment (Ex: bank-transfer)
Datum: record date-time
Invoicenum: the number of the invoice
~~Invoiceid: deprecated~~
Invoicefrom: billing period begin time
Invoiceuntill: billing period endtime
Invoicegenerated: invoice print time (optional)
Invoicesent: invoice sent time (optional)
Completitiondate: completition date
Duedate: payment due date
Language: invoice language
Vat: VAT percent
VatValue: sum of VAT
InvoicePrice_Net: final net price
InvoicePrice: final price
PaymentReceived: date-time of the payment
Debt: sum of all debt
Pending: all sum before due date
Comment: invoice comment
InvoiceImage: saved ivoice file

tb_invoice_entries:
Id: autoincrement database primary key
Invoiceid: foreign key to tb_invoices
datum: record insert date
description: product description with code
fromdate: billing period if applied
todate: billing period if applied
Ammount: ammount
AmmountName: name of the amount (minute)
AmmountPrice: unit price (net)

The emmitent (company) settings are stored in the tb_**providers** table. Only one company entry can be stored (conform to Hungarian laws). *Once the company details are inserted to the database, the company name cannot be changed anymore.*

4.5.9. Environment variables

Creditelapseunit: prepaid credits will be elapsed automatically after this period is elapsed. Creditelapseunit means the credit amount for 1 day. For example if you set it to 40 and the user will buy 5000 ft credit, than it will elapse after 4 month
Currency: the currency type is loaded from the "currency" global configuration or from the billed user currency setting.
The price setup currency settings also can affect the currency settings.
Currency conversions: if the pricelist is not in the native currency format (set by the "currency" global config option), than the server can convert to it automatically based on tb_currencies. You can change the conversion rates from MManage -> Billing -> Currency converter
Language: the invoice language can be controlled from the invoice form.
PDF Printer and delay: set this on MManage -> Menu-> Options
Time format: if to separe duration values to day/hour/min/sec or display only in seconds. (MManage -> Menu-> Options)
Money precision and rounding and separator are stored in the tb_currency_precizion table, accessible from the Billing form.
MINSPEACHLENONROUTE: the minimum calculated max speechlength when the call will be routed

4.5.10. Payments

There are various built-in prepaid and postpaid payment method implemented. Payments are tracked in the tb_invoices table and can be queried later for statistics and reports (Billing -> Reports form)
All **credit changes** for prepaid users should be logged in this user. Never modify the credit directly. Use the "Modify" button from Users and device -> Billing page if direct modification is required.
Invoices for postpaid user can be generated from the Billing -> Invoices
Chargecards can be generated from Billing -> Pincodes
CallingCards:
There is a special user type called "callingcards" but any usual user can act as a calling card.
Users can access the system via Web or IVR typing a pincode. The pincode will represent the "pincode" column from the user table or the username+password combination for enduser or only the username field for calling cards.
PayPal: direct or indirect handling of PayPal payments are supported. Search for "paypal" in the global configuration to setup. See the FAQ for the details about the configuration.
Your users are allowed to use **e-payment** and to pay directly with their **credit card**. Most of the available merchant gateways are supported.
Credit Card and eCheck processing support for every major Internet Payment Gateway using secure data communications using up to 128-bit SSL encryption and Digital Certificates.
The Credit Card validity checks decrease expenses that result from attempting to authorize invalid credit cards.

The current list of supported payment gateways include:

3DSI EC-Linx	www.3dsi.com
ACH Payments	www.ach-payments.com
Authorize.Net	www.authorize.net
Bank Of America	www.bankofamerica.com
BeanStream	www.beanstream.com
Chase Merchant Services	www.chase.com
Concord EFSNET	www.concordefsnet.com
CyberCash	www.cybercash.net
Cyber Source	www.cybersource.com
DPI Link	www.dpicorp.com
ECHOnline	www.echo-inc.com
ECX QuickCommerce	www.ecx.com
eProcessing	www.eProcessingNetwork.com
eWay	www.eway.com.au
Fast Transact	www.fasttransact.com
FirstData / Cardservice Int.	www.firstdata.com
goEmerchant	www.goemerchant.com
GoRealTime (Full-pass)	www.gorealtime.com
Innovative Gateway	www.innovativegateway.com
Intellipay ExpertLink	www.intellipay.com
Iongate	www.iongate.com
iTransact RediCharge HTML	www.itransact.com
LinkPoint	www.linkpoint.com
Merchant Anywhere	www.merchantanywhere.com
Merchant Partners	www.merchantpartners.com
Moneris	www.moneris.com
MPCS Weblink	www.merchantcommerce.net
NetBilling	www.netbilling.com
Network Merchants	www.networkmerchants.com
NexCommerce	www.thompsonmerchant.com
NOVA's My Virtual Merchant	www.myvirtualmerchant.com

NOVA's Viaklix	www.viaklix.com
OGONE	www.ogone.be
Optimal Payments	www.optimalpayments.com
PayFlow Link	www.paypal.com
PayFlow Pro	www.paypal.com
PayFuse - First National MS	www.firstnationalmerchants.com
Paygea	www.paygea.com
PayJunction Trinity	www.payjunction.com
Paymentech - Orbital	www.paymentech.com
Payment Express	www.paymentexpress.com
Payments Gateway	www.paymentsgateway.com
Payready Link	www.payready.net
PayStream	www.paystream.com.au
Planet Payment	www.planetpayment.com
Plug 'n Pay	www.plugnpay.com
PRIGate	www.paymentresource.com
Protx	www.protx.com
PSIGate	www.psigate.com
RTWare WebLink	www.rtware.net
SECPay	www.secpay.com
SecurePay	www.securepay.com
SkipJack	www.skipjack.com
Sterling	www.sterlingpayment.com
SurePay / YourPay	www.surepay.com
TransFirst eLink	www.transfirst.com
TrustCommerce	www.trustcommerce.com
USA ePay	www.usaepay.com
uSight	gateway.usight.com
Verifi	www.verifi.com
Verisign PayFlow Pro	www.verisign.com
WorldPay Select Junior Invisible	www.worldpay.com
YourPay	www.yourpay.com

and more ...

Search for epayment in the global settings for the configuration details.

To enable the E-Payment module follow these steps:
-1. Install the epayment module: EPaymentIntegrator (request from support if you haven't received)
-2. Register the e-payment module
-3. Set the epayment_xxxx settings properly in the global configuration

E-Payments can be initiated from softphone, web portal or the module can be accessed by any external application using the console or the database API.
Any other third party payment method can be integarated.

4.5.11. Resellers

If the "resellerbilling" global config option is set, than reseller cdr records are stored in the tb_cdrresellers and billed accordingly.
To define a "base tariff" for the reseller the "Is Public" option is used from the "Price Settings" form (usually applied to an "enduser cost" packet.)
This will be the prices that will have to be paid by resellers to the service owner.
Reseller can create their base tariff (by setting the "resellerid" in tb_billentries) usually from a web portal. Multiple packets are allowed and packets can be assigned to other users or resellers directly from web (in the same way like on the "Users and Devices" form -> "Billing" tab -> Billing packet setting.
Resellers usually will create their own price lists by cloning an existing list or their base tariff list.
For early billing set the reseller stage field to 9.

Top reseller statistics can be viewed on the "Statistics" form by checking the "OC" and the "PR"/"PFR" fields.

4.5.12. Promotions

The following features can be used to use various promotions to (new) users:

- First X seconds are free
  You might create packets when the first few seconds are not billed.
  Just use the "Free amount" option on the Price Setup form for this
- Call X direction for only … cent
  Just setup a separate packet with lowered prices for this.
  Use only the directions (prefixes) you wish to promote.
- 10 USD free to direction X
  Set the "packetcredit" field in tb_users to 10

Create a special packet with the desired directions (prefixes). Set the "isforpacket" field to 1 for this packet(s)
With these settings the cost of the call is calculated with this special packet(s) and the cost will be deducted from the "packetcredit" field instead of the real "credit" field.

- You can offer some directions for the users where they can call Y minutes for free

  The followings fields have to be used for this from the tb_users table:

  - freeminutes: free minutes included
  - freesms: number of free sms messages included
  - freeminutesleft: remaining free seconds! (this field will be modified by the server. No more free minutes when reach to 0)
  - freesmsleft: remaining free sms (this field will be modified by the server. No more free sms when reach to 0)
  - freedirections: numbers separated by comma or a prefix where the free minutes or sms can be used (it can be also a whole number). Empty value for this allows free calls to all directions.
  - freedays: set to a positive value if you wish the free minutes to be reset after this number of days (recurring free minutes). Set to 0 to disable reset. (By reset we mean that the freeminutesleft will be set to freeminutes)
  - freebegindt: freedays begin date-time

  For this feature to be used you might have to set the "checkfreeminutes" global config option to 2. You might also set the "freedirections" global config option to restrict the directions that can be allowed by the tb_users. Freedirections field (if you let this field to be modified freely by the users, for example to select one preferred number or destination prefix)

  Example SQL to add 100 free minutes ad 30 SMS for a set of users to prefix 4411:

  ```
  update tb_users set freeminutes = 100, freesms = 30, freeminutesleft=100*60, freesmsleft=30, freedirections ='4411', freebegindt=getdate(),freedays=31 where id in (X,Y,Z)
  ```

  To add 10 free minutes for new users, just set the default value for the "freeminutes" field to 10 in the database.

- Offer some free minutes for all new users (restricted by user IP)

  There is another possibility by restricting the maximum number of calls or call duration from a source address by the following global settings:

  - maxcallperip: max number of calls from the same IP
  - maxdurationperip: max call duration from the same IP
  - maxcostperip: max call cost per IP
  - maxcallperperiod: the period for the above (1.0 means one day)
  - maxcallperuserid: the above will be set to this user only (set to -1 for all users)

  Additionally to IP restriction you should set as much other restrictions as possible to restrict your server attack surface, such restricting the usage to one user (maxcallperuserid), set user as prepaid, set max lines and daily/monthly limits on the account as other best practices discussed in this documentation.

4.5.13. Fraud protection

See the "Security and account limiting" section,

4.5.14. Notes

Call forward billing: 2 cdr record will be generated. A->B and B->C
Call forward from IVR: one cdr will be generated. Whether we charge the call to the IVR or only bill the forwarded call can be controlled by "resetdurationonfwd"
Call transfer by SIP signaling: the second call is completely different from the first call. Billing goes normally (2 calls)
Call transfer with dtmf (*5*): only one call leg is billed
Conference with dtmf (*1*): separate cdr will be generated for all call legs
Conference by sip: technically separate calls. Will be billed normally (2 cdr)

Call forwarding from IVR (2-leg calls):

CDR's generated based on "ivrbilling" global and user setting: 0: one CDR including the forwarded call, 1: load duration only from forwarded call, 2: generate 2 CDR records (A leg + B leg), 3=both merged,4=merged with short a-leg,5=only b-leg billing if call is connected

- ivrbilling is 0: (server side) 1 CDR will be generated with total client call duration. The billing will be done after the final called user (the IVR accessnumber when the call was not forwarded. Otherwise the final destination number)
- ivrbilling is 1: (client side) 1 cdr will be generated. The call duration will be set after B-leg call duration and billed accordingly
- ivrbilling is 2: (both) 2 (or more) cdr will be generated (when there was call forwarding action). The 2 cdr record can be billed separately after different billing tables
- ivrbilling is 3: (both merged) 1 cdr will be generated, but the enduserprice can be loaded from different billing tables (2-leg merged)
- ivrbilling is 4: (both merged with short a-leg) 1 cdr will be generated, but the enduserprice can be loaded from different billing tables (2-leg merged). The A leg duration is shortened (only the time spent with IVR until the call forward action)
- ivrbilling is 5: (server side if connected –mostly the same like ivrbilling 0) 1 CDR will be generated. If the call was not connected then all duration will be billed (you can setup different billing for these calls by marking the entry as "is ivr call" and set the "called" to the access number. If the call is connected, then the B-leg will be billed (possibly after a different billing packet)

4.6. WebRTC

The Mizu VoIP server has support for WebRTC since version 7.4 (including websocket SIP signaling and DTLS/SRTP encoding/decoding)
The server is capable to handle WebRTC-WebRTC routing or transparent SIP to WebRTC or WebRTC to SIP conversion.

The WebRTC module description have been moved to a separate document:
https://www.mizu-voip.com/Portals/0/Files/WebRTC_SIP_Gateway_Doc.pdf

The above document focuses on the MRTC gateway, but most of it applies to the VoIP server as well.

4.7. Push notifications

The Push module description have been moved to a separate document:
https://www.mizu-voip.com/Portals/0/Files/VOIP_Push_notifications.pdf

The above document focuses on the MPUSH gateway, but most of it applies to the VoIP server as well.

4.8. Other –MManage

Other not so important MManage modules are described below.

4.8.1. Configurations

Global system configurations.
Basic configuration are vital for the system to run correctly.

A list of settings are described below. New versions usually adds also more configuration options which you can find from the Configurations form.

Category	**Key**	**Description**
CallCenter	allowdbcalls	Allow calls from database in MAgent
CallCenter	allowmanualcalls	Allow manual calls from MAgent
CallCenter	allowopcampchange	Allow operators to change its campaign from MAgent
CallCenter	autoformname	Type of AutoCall GUI to load
CallCenter	callbackautorecall	1= schedule for recall if number is in campaign
CallCenter	callbackhandling	handling incoming calls: 0=dropp all
CallCenter	callbackivr	play special messgae (callback.wav) if no operator found or ring timeout expired
CallCenter	callbacknumber	A number for calls. For example the predective dialer will use this number
CallCenter	callbackringtimeout	ring timeout on callback (after than play ivr message if set) defrecallmin
CallCenter	callbackroutenumber	number to be dialed on incoming calls when callbackhandling is 1
CallCenter	CALLCENTERPORT	callcenter tcp port number (for requesting new clients)
CallCenter	callmaxring	max ring time in sec

CallCenter	callmaxwait	max worktime for operators between calls
CallCenter	callnumbertype	Preference order of numbers (if more than one number exists for a client): 0=first try landline
CallCenter	callretryival	seconds to wait untill to redial the client
CallCenter	ccorder	client call order: 0=database id order
callcenter	defrecallmin	recall data-time will be shown with the specified minute in MAgent
CallCenter	desireddropprate	optimal percent of calls wich cannot be assigned to operators when in predective
callcenter	finishvoice	default file when set to "finish with voice" in scripts
CallCenter	handlingincoming	0=not handled
CallCenter	maxcallatonce	max number of calls in one round when in predective (error guard)
CallCenter	maxcallsperminute	max new call attempts/minute when in predective (error guard)
CallCenter	maxcalltrycount	max attemt of calls for a client
CallCenter	maxrecallafter	client can be recalled in this interval
CallCenter	maxrecallafterall	client can be recalled in this interval by any operator
CallCenter	maxrecallbefore	client can be recalled before the specified time
callcenter	maxrecallmin	restriction of the recall date-time input in MAgent
CallCenter	maxrecalltrycount	max attemt of REcalls for a client
CallCenter	mobileratio_08_12	percent of mobile calls in the specified period [1-100]
callcenter	mobileratio_12_16	percent of mobile calls in the specified period [1-100]
CallCenter	mobileratio_16_20	percent of mobile calls in the specified period [1-100]
CallCenter	mobileratio_20_08	percent of mobile calls in the specified period [1-100]
CallCenter	mobileratio_weekend	percent of mobile calls in the specified period [1-100]
CallCenter	predectivecheckival	controlls the speed of the predective dialer thread -advanced technical setting
CallCenter	predectivecorrection	correction of precalculated success ratio statistics in predective. for example if we set it to 80
CallCenter	predectivecutnofreeop	disconnect pending predective calls if no more operator waiting
CallCenter	predectivedial	dialing mode: 0=simple MAgent requests
CallCenter	predectivelogging	details of predective logs (0=no logs
CallCenter	predectivemaxsucccalls	max predective calls in the same time (check in calllist)
CallCenter	predectivemaxsuccmobilecalls	max predective mobile calls in the same time (check in calllist)
callcenter	presentationmode	0=no presentations
CallCenter	quotastatrecalcival	how often the quotas will be recalculated
callcenter	recallonlyinsamecampaign	call only with its own campaign (no recalls from other campaigns)

CallCenter	recallrescheduleival	if call continue to fail
CallCenter	recallrescheduleivalfirst	if call failed for the first time
CallCenter	recallrestrictions	recall mode: 0=give to the same operator
CallCenter	savemode	level of saving data - 1=automatikus mentes a kovetkezo kerdesre ugraskor
CallCenter	statival	rebuild predective statistics interwal (-2 = automatic)
CallCenter	stopwrongcdrc	pause predective if the ASR is too low (error guard)
CallCenter	waitforpredective	max time to wait for a free operator when a predective call is connected
CallCenter	waitifnotaccepted	max time to wait in MAgent if the client is not "accepted"
CallCenter	waitifnotconnected	wait after calls even if was not connected (operator worktime)
gatekeeper	h323debuglog	write h323 gk log to logfile
license	CAN_alert	default license (will have no effect if you change it here)
license	CAN_billing	default license (will have no effect if you change it here)
license	CAN_failover	default license (will have no effect if you change it here)
license	CAN_filtering	default license (will have no effect if you change it here)
license	CAN_gsmextra	default license (will have no effect if you change it here)
license	CAN_hash323	default license (will have no effect if you change it here)
license	CAN_hassimbank	default license (will have no effect if you change it here)
license	CAN_hassimplatform	default license (will have no effect if you change it here)
license	CAN_hassip	default license (will have no effect if you change it here)
license	CAN_recharge	default license (will have no effect if you change it here)
license	CAN_runsipproxy	default license (will have no effect if you change it here)
license	CAN_sipextra	default license (will have no effect if you change it here)
license	FULLRIGHTS	default license (will have no effect if you change it here)
license	lic_isset	default license (will have no effect if you change it here)
license	LICENSEMAXMONTH	default license (will have no effect if you change it here)
license	LICENSEMAXYEAR	default license (will have no effect if you change it here)
license	MAXALLUSERS	default license (will have no effect if you change it here)
license	MAXCCALS	default license (will have no effect if you change it here)
license	MAXCHANNELS	default license (will have no effect if you change it here)
license	MAXGATEWAYS	default license (will have no effect if you change it here)
license	MAXSIPUSERS	default license (will have no effect if you change it here)

license	MAXSL	default license (will have no effect if you change it here)
license	MAXTRAFFICSENDERS	default license (will have no effect if you change it here)
license	SRVVERSION	default license (will have no effect if you change it here)
licensecfg	hasalerting	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hasbilling	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hascallcenterin	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hascallcenterout	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hasextra	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hasfailover	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hasfiltering	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hasgsmextra	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hash323	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hasrecharge	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hassimbank	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hassimplatform	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	hassip	modules to load (has effect only if not prohibited by builtin license restriction)
licensecfg	maxallusers	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxcallspermin	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxccalls	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxccallsblock	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxchannels	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxgateways	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxgsmgateways	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxivrspeechlen	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxregistrations	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxsessionspeechlen	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxsipusers	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxsl	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxtagents	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	maxtrafficsenders	limitations (has effect only if not prohibited by builtin license restriction)
licensecfg	runsipproxy	run h323 - sip translator

settings	adminport	adminport
settings	alertonlowdiskspace	alert on low disc space
settings	aliasrouting	allow lrq routing
settings	allownumbersendback	allow to route back the call to the caller
settings	autodetectlocalip	automatically overwrite the localip value if set to true
settings	autoholiday	holiday routing and billing treated as Sunday
settings	billshortnumlength	number that are smaller than this value will not be billed
settings	bindip	bind to this ip (for multihomed servers or if we run multiple serers on the same maschine)
settings	boostonfirstcall	if to start with low priority and boost it when the first call arrives
settings	brs_lcr	routing algoritm: 0=not used
settings	buildloadstatistics	tb_loadstatistics
settings	canaoutosaveinifiles	if we can save unsaved items
settings	canrestartformalfunctions	0=cannot restart
settings	cfg_block711	if g711 (PCMU
settings	checkblacklist	0=no check
settings	checkcallerids	wich calls are to be checked on the selfcheck thread (useful if you have wrong traffic senders)
settings	checkcputime	if cputime is constantly high
settings	checkcredirights	if to check the owner of the sim and chargecards
settings	checkgkcdrs	if to check cdr records from the gk statusport
settings	checkgsmnumlen	if to allow incoming number only with this size
settings	checkknownnumbersex	0 = no
settings	checklocalnumberpx	local endusers prefix to check (to not route to other servers). if emty
settings	checklocalnumbers	check local endusers on routing
settings	checkmaxlines	check max used lines to client and partner
settings	checkmaxlinetb	extended check for max lines (tb_maxlinep)
settings	checkmaxnumlen	max called number length allowed
settings	checkminnumlen	min called number length allowed
settings	checknumlen	if to check the incoming number len //14
settings	checknumport	check number portability
settings	checknumportpx	check number portability for this prefixes
settings	checkpacketfailovering	0=don't check

settings	checkprefixes	check asr/acd for alerting and wachdog only for this prefixes. must be in the following format: 'xxx'
settings	checkprefixroules	set to 1 to check additional prefix rules. by default only H323 ip prefix rules are applied
settings	checkrouterrights	callprefixrights and partner binding
settings	checksss	check special numbers
settings	connectondisccode	if to connect the sipcall before to play the disconnect reason. 0=not connect
settings	country	used in number normalizations
settings	countryprefix	used on routing
settings	creditcheckforpostp	check credit for postpaid users too (max limit)
settings	creditcheckforts	check credit for traffic senders too
settings	currency	local currency
settings	dailymainttaskhour	when to perform daily maintenance tasks
settings	dbdelbackupdir1	additional backup directory to cleenup
settings	dbdelbackupdir2	additional backup directory to cleenup
settings	dbdelbackupdir3	additional backup directory to cleenup
settings	dbloglevel	db server loglevel (0=only errors to monitor
settings	dbmaint_backupdbdir	database backup directory
settings	dbmaint_backupdbnetworkdir	database backup directory path (if the database engine is located on a remote server)
settings	dbmaint_backuplevel	0=no backups
settings	dbmaint_backuptables	backup cdr records and other tables to xxx_backup: 0=no
settings	dbmaint_do	do database maintanance
settings	dbmaint_removecdrs	remove cdr records after x days
settings	dbmaint_removelogs	remove logfiles after x days
settings	dbmaint_removeother	remove other tables records after x days
settings	dbtimeout	database query timeout
settings	defaultenduserprice	if no price entry found
settings	defaultproviderprice	if no price entry found
settings	defcallerid	default caller id for cdr records if cannot be determined exactly
settings	deldbbackup	delete old backup files after this day elapsed
settings	deleteoldlogfiles	delete older logfiles than the specified day (set to 0 to disable)
settings	emailbatchwait	bulk email sender wait interval
settings	emailfromaddr	default email config

settings	emailfromname	default email config
settings	emailhost	smtp server used for alerting
settings	emailsubject	default email config
settings	emailuser	smtp username used for alerting
settings	emergencydir	route emergency calls to this gateway (user id)
settings	enablefirewall	enable/disable builtin firewall and dos attack filtering
settings	enforcestrongauth	enforce authorization and strong passwords
settings	estimatedspeachlenoncard	sim card config
settings	eveningbegin	evening begins at that hour. used for common time intervalls
settings	fakegwalias	will route the wrong calls here. set to FBACKUPGW if needed
settings	faxfromaddr	fax sender configuration (email to fax server)
settings	faxfromname	fax sender configuration (email to fax server)
settings	faxhost	fax sender configuration (email to fax server)
settings	faxnormalize	fax sender configuration (email to fax server)
settings	faxsubject	fax sender configuration (email to fax server)
settings	faxsuffix	fax sender configuration (email to fax server)
settings	faxuser	fax sender configuration (email to fax server)
settings	fileloglevel	file server loglevel (0=only errors to monitor
settings	filetransferbufflen	fileserver buffer length
settings	filetransfertick	fileserver speed
settings	freenumlen	number length that can be called free of charge
settings	gkcommand	how to start the h323 gatekeeper
settings	gkstoptick	gk setting
settings	globalcdrid	server generated. don't touch
settings	gmtdiff	the difference to gmt (useful for sip date header)
settings	gsminccallerip	convert incoming caller ip from gsm gateways when forwarding to a support phone (incall action in gsm gateways is
settings	incduration	to increase cdr duration. 1=small inc ~ 1%
settings	internal_endusercost	enduser price for internal calls (between endusers)
settings	internal_providercost	provider price for internal calls (between endusers)
settings	InternalIP	sipserver internal ip (interface to clinets)
settings	keepbackuprecorded	days to keep voice records in the backup directory

settings	keeprecorded	days to keep voice records
settings	LocalIP	sipserver external ip
settings	loglevel	other server loglevel (0=only errors to monitor
settings	lognofreecardc	list free card data when no route found
settings	logsqlcommands	trace sql commands (log)
settings	logtodb	trace to database (log)
settings	logtofile	trace to file (log)
settings	maxdisableonss	reserver sim capacity config
settings	maxgkmemoryutilization	max memory for gatekeeper in KB (restart if exceed)
settings	maxloglisten	max log message quee length
settings	maxmemoryutilization	max memory utilization in KB (restart if exceed)
settings	maxnocdrmin	will take the correspondig action if no cdr record found in the last "maxnocdrmin" minute. set to 0 to disable checks
settings	maxnogkcallmin	will take the correspondig action if no new call
settings	maxnologival	will take the correspondig action if no log entry found in the last "maxnologival" minute. set to 0 to disable checks
settings	maxnosip2h323callmin	will take the correspondig action if no sip2h323 call
settings	maxpriceperminute	will alarm if providerprice is bigger
settings	maxrecdiff	recorded voice stereo sync in msec
settings	maxrecdiff2	recorded voice stereo sync in msec
settings	maxroutereqpermin	allow only "maxroutereqpermin" routing request/minute
settings	maxudpselect	max socket on select (set to -2 to autoconfigure. -1 means no limits)
settings	minacd	min acl value
settings	minactivegateways	will take the correspondig action if not enough gateways
settings	minactivesims	will take the correspondig action if not enough channels
settings	minasr	min asr allowed. will alert/reset if lower
settings	minblockcallcount	mincallcount to check in periodic blacklist refresh
settings	mincdrcount20min	minimum number of calls/20 min. restart if lower
settings	mincreditoncharge	general credit limits
settings	mincreditonroute	credit limits
settings	minfreechargecards	min free sim charge card
settings	minlinetoprefix	min line to the predefined prefix
settings	minlogdelay	minimum delay between writing two log messages in msec

settings	minmemoryutilization	will restart on offpeak if exceed
settings	minprofit	alerting threshold
settings	minprofitpercent	alert config
settings	minsubsecventcallbegin	wait at least minsubsecventcallbegin when route to the same channel
settings	mobileprefixes	mobile number prefixes
settings	monitorport	default monitor port
settings	nightbegin	night begins at that hour. used for common time intervals
settings	normalizenumbers	0=not at all
settings	numfiltering	blacklist/whitelist filter (0=don't filter
settings	onlyroutealias	you can set a gateway here. all traffic will be routed to that gw
settings	onlyroutesim	you can set a simid here. all traffic will be routed to that sim
settings	peaktimebegin	peaktime settings for billing and routing
settings	peaktimebegintr	peaktime settings for various operation
settings	peaktimeend	peaktime settings for billing and routing
settings	peaktimeendtr	peaktime settings for various operation
settings	ppriority	0=low
settings	removetrailinghash	remove # when routing
settings	restartatnight	set to true if you want a reset every night
settings	restartpcatfirst	restart the pc immediately on error (don't restart the sw)
settings	rotatelogfile	create separate logfiles for every day
settings	rrr_black	blacklist q931 disconnect code. defaults to ResourceUnavailable = 47
settings	rrr_denied	access denied q931 disconnect code. defaults to ChannelUnacceptable = 6
settings	rrr_desterr	wrong destination q931 disconnect code. defaults to DestinationOutOfOrder = 27
settings	rrr_err	routing error q931 disconnect code. defaults to TemporaryFailure = 41
settings	rrr_nogw	no capacity q931 disconnect code. defaults to NoCircuitChannelAvailable = 34
settings	rrr_tomany	too many wrong request q931 disconnect code. defaults to Congestion = 42
settings	rrr_wrongnum	wrong B number q931 disconnect code. defaults to InvalidNumberFormat = 28
settings	runfakegw	if to run local fake h323 gw to offload exceeding traffic
settings	runstatistics	long havy cdr statistics
settings	salescomissionfromprofit	The defined commission percent will be calculated for the profit or for the enduserprice
settings	sendmailalert	if to send alerts on critial errors (pease configure the emailalertX settings)

settings	serverftpdailydir	create separate directory for recorded voices daily
settings	serverftpdir	used for inifiles
settings	serverftpvoice	where to store recorded audio
settings	servername	server name (will appear in reports
settings	shortnum_endusercost	enduser cost for short numbers
settings	shortnum_providercost	provider cost for short numbers
settings	sipcommand	how to launch the local h323-sip server
settings	SIPH323GWID	sip to h323 protocoll conversion will be done using this gateway or module
settings	siph323gwid2	sip2h323 user id (if -1 than will load automatically)
settings	skipsqlerrors	if to skip sql errors (not throw)
settings	storecdrcomments	sore comment to cdr records. 0=no
settings	storeduplicatecdr	store separate cdr record for sip 2 h323 conversions
settings	usedefaultdisccodes	don't use customized disconnect codes
settings	usedelayedupdate	sql upfates in separate thread
settings	usewrongnumfilter	block (nearly) subsecvent wrong callednumbers (0=don't block
settings	voicebackupdir	where to store a backup of recorded conversations
settings	weekendispeak	peaktime settings for billing and routing
settings	weekendispeaktr	peaktime settings for various operation
settings	wrongnumcache	remembered callednumbers. used if usewrongnumfilter is not 0
settings	wrongnumdropcount	drop call if this number of failed calls found in cache. . used if usewrongnumfilter is not 0
SIMAllocation	simallocivaltype	simbank config
SIMAllocation	simallocsendival	simbank config
SIMAllocation	simalloctimebefore	simbank config
SimPlatform	allowlastsim	specify to allow two subsequent calls to be routed on the same simcard
SimPlatform	allowlastwrongnum	specify to allow subsequent wrong number
SimPlatform	allowoldstat	allow old status messages
SimPlatform	autosimalloc	automatically allocate simcards
SimPlatform	brsforgsm	check BRS for gsm gateways too
SimPlatform	checkcredirights	sim and chargecard rights
SimPlatform	checkcreditandcharge	if to run the creditcheck thread
SimPlatform	checksimsdisable	automatic simcard prioritization based on ASR and ACD

SimPlatform	checktpercek	allow tpercek exception in routing
SimPlatform	checkwestelcards	Hungarian specific
SimPlatform	convertsimcreditcurrency	convert incoming credit message to native currency (defined by "curerency" config value)
SimPlatform	convertsimcredittonovat	convert oncoming credit message to not include VAT values
SimPlatform	creditchecktimeival	run the creditcheck thread for every X interval
SimPlatform	creditunit	usefull in credit checks. equilavent with 600 ft. less if you want more precise credit calculations
SimPlatform	defaultsimpacket	use the default packet for new simcards (simcards without a packet)
SimPlatform	enforcegwname	accept only gateways with valid name. the 'GW' suffix must be present in names
SimPlatform	estimatedspeachlenoncard	needed for routing
SimPlatform	fakesmscount	fake sms messages by sim / month
SimPlatform	fieldstrengthstatistics	build fieldstrengthstatistics
SimPlatform	fnosamecountyc	used for fake sms and calls. every fnosamecountyc sms/calls will go to some other country
SimPlatform	fnosamepacketc	used for fake sms and calls. every fnosamepacketc sms/calls will go to some other simpacket
SimPlatform	fnosameprovidc	used for fake sms and calls. every fnosameprovidc sms/calls will go to some other provider
SimPlatform	gkpname	gatekeeper name
SimPlatform	gkrstifnotconnected	restart the gk if not connected
SimPlatform	gsmgwport	port for the gsm gateways
SimPlatform	gsminccalled	forward incoming call from gsm gateway to this number
SimPlatform	gsminccaller	convert incoming caller number from gsm gateways when forwarding to a support phone
SimPlatform	gsminccallerip	gsminccallerip
SimPlatform	gsminfromip	fromip
SimPlatform	gwstatistics	to build the table or not
SimPlatform	incomigcalls	simulate incoming calls on simcards. 0=no
SimPlatform	incomingcallcount	number of incoming call/month by simcard (if the incomigcalls options is set to 1 or 2)
SimPlatform	lowercredit	sss
SimPlatform	maxallowedcredit	only values below are valid (when received in sms or dtmf)
SimPlatform	maxflysims	max flying simcards
SimPlatform	maxsimpreconfig	max credit rec delta
SimPlatform	maxsimpriceonalloc	max accepted price
SimPlatform	mincreditoncharge	min credit
SimPlatform	minfreelines	will take the correspondig action if not enough free line found

SimPlatform	minsimlastrectime	elapsed minutes when the server decide that the channel is offline
SimPlatform	onlyroutealias	route all calls to this gateway
SimPlatform	onlyroutesim	route all calls to this simcard
SimPlatform	reservedef	reserver capacity
SimPlatform	reserveforh323	reserve capacity for h323. reservations will be disabled if less than 1
SimPlatform	reserveforsip	reserve capacity for sip . reservations will be disabled if less than 1
SimPlatform	sendfakesms	send sms messages between simcards
SimPlatform	simalloccangominus	we may activate sims that have negative values
SimPlatform	simcreditdiv	reported credit must be divided by this number
SimPlatform	succhargetime	minimum time between succesive charge try on the same simcard in minutes
SIPSettings	ABSOLUTETIMEOUT	max session time (call duration + setup time + clearing time) in seconds
SipSettings	addcontentdispozition	0=no;1=optional;2=required
SIPSettings	allowcallunregistered	allow to call before registered (terminals)
SIPSettings	allowdiscmessage	allow disconect reason voice playback
SipSettings	allowlist	INVITE
SIPSettings	CanAcceptLocalIp	Can call from 127.0.0.1
SipSettings	cancutsipnumbers	packet dialplan for sipnumbers
SipSettings	canmove	0=not allowed
SipSettings	checknomedia	disconnect calls on rtp disconnect
SIPSettings	COMPANYNAME	this will apear in the sip signaling
SipSettings	def_max_sessiontimer	sip session-timer config
SipSettings	def_mid_sessiontimer	sip session-timer config
SipSettings	def_min_sessiontimer	sip session-timer config
SipSettings	domainnames	registrar domainnames (used for inter-domain rerouting)
SipSettings	eventlist	refer
SipSettings	forwardretrytimer	ivr forward retry interval
SipSettings	fwdtootherdomains	0=no
SIPSettings	fwdunknownheaders	forward unknown sip headers
SIPSettings	HasInternalAccess	accept from 192.168... or 10.0....
SIPSettings	identityrwmode	0=no rewrite
SIPSettings	IDLETIMEOUT	used for various session timers

SIPSettings	im_billinguserid	used for instant messaging
SIPSettings	im_parentid	used for instant messaging
SIPSettings	lastlocalsdpport	used in terminals
SIPSettings	loadcallednumberfromto	load the called number from sip to instead from the sip first line (URI)
SIPSettings	localclientport	useful for 2 port configurations
SIPSettings	LocalDomain	sipserver domainname
SIPSettings	localinternaldomain	sipserver internal domainname
SIPSettings	LocalPort	sipserver listen port
SipSettings	logsipmsgexchange	store the sip message headers in the cdr comment
SIPSettings	MAINTIMERIVAL	sip background process timer. used for garbage collections mainly
SIPSettings	MAXEPCOUNTTRESHOLD	maximum number of registered endpoint (it may be limited by license too)
SIPSettings	MAXH323GKCDRCACHE	this must at least the maximum h323-h323 simultaneous call number
SIPSettings	maxreroute	max number rute retry
SipSettings	MaxRTP	rtp port range begin for sip
SIPSettings	MAXSPEACHLEN	max allowed call duration in sec
SipSettings	maxstatchangepermin	max allowed enduser status changes/60 sec (slower if exceed)
SIPSettings	MAXSUBSMSGCOUNT	max subsequent messages before block
SIPSettings	MAXSUBSMSGPERIOD	max subsequent messages before block are checked for this interval (sec)
SIPSettings	MAXWRONGMSGALLOWED	dos attack protection
SIPSettings	MEDIATIMEOUT	will disconnect if the media disappears
SIPSettings	MEDIATIMEOUTSTART	will disconnect if no media detected
SIPSettings	MINRESENDIVAL	sip udp resend timer (T1) in msec
SipSettings	MinRTP	rtp port range begin for sip
SIPSettings	MINSPEACHLENONROUTE	minimum remained speechlength for the caller when the router will still route the call
SIPSettings	MINUSERCREDITONROUTE	minimum credit for the caller when the router will still route the call
SIPSettings	MINUSERNAMELEN	minimum accepted username length
SIPSettings	PRODUCTNAME	the name of the product. this will apear in the sip signaling
SipSettings	propersipcomments	set to true if you want personalized sip headers
SIPSettings	REBUILDREGCLIENTS	usually the same as maxspeachlen
SIPSettings	REENABLEDOSBLOCKED	reenable blocked endpoints after this interval. defaults to 12 hour
SIPSettings	REGISTERIVAL	upper registration interval in msec. defaults to 40 min

SIPSettings	RELOADPROXYLISTIVAL	reload proxies from the config
SIPSettings	REPOPUPULATEFDSETIVAL	used for rtp routing
SipSettings	REPOPUPULATEFDSETIVAL_MAIN	used for main routing
SIPSettings	rerouteon	rerouting behaviour: 0=disabled 1=onlyifset (on busy
SipSettings	resolvedns	resolve uri domain names
SipSettings	ringtimeout	sip calls ring timeout
SIPSettings	routepresence	route subscribes
SipSettings	rtpsendonlytorec	send the rtp only to the rec address
SipSettings	rtpwritefirst	send a rtp packet after connect (to open NAT)
SipSettings	sdpalthandling	0=not handled
SipSettings	sessiontimer	0: don’t use
SipSettings	sipmsgresendcount	resend sip message count
SipSettings	sipmsgresendival	sip message resend timer
SipSettings	sipsendtodefport	try to send to port 5060 too
SipSettings	statussaveival	minutes. used when predective is active
SipSettings	supportlist	privacy
SIPSettings	traceep1	user id
SIPSettings	traceep2	user id
SipSettings	udpkeepalive	send keepalive messages
SIPSettings	udppriority	rtp thread priority: 1=normal
SipSettings	upperexpire	register expire
SIPSettings	usedateheader	send date to user agents
SIPSettings	userofflinemin	enduser will be considered offline if no register or invite for this period
supervisor	canrestartformalfunctions	if supervisor can do restarts. 0
supervisor	checkcallerids	check only this cdr records
supervisor	checkprefixes	check only this cdr records
supervisor	maxnocdrmin	restart if no cdr records
supervisor	minacd	min treshold
supervisor	minacl	min treshold
supervisor	minasr	min treshold
supervisor	peaktimeendtr	peaktime end hour

supervisor	restartatnight	if to restart at every night
supervisor	restartpcatfirst	don't restrart the service. restart the pc immediatly
supervisor	trafficammount	0=after setup
supervisor	weekendispeak	treat weekend as peaktime (same traffic ammount)

Other config options:
Show/hide child's and spouse from accept action:
„showspouoseandchild:

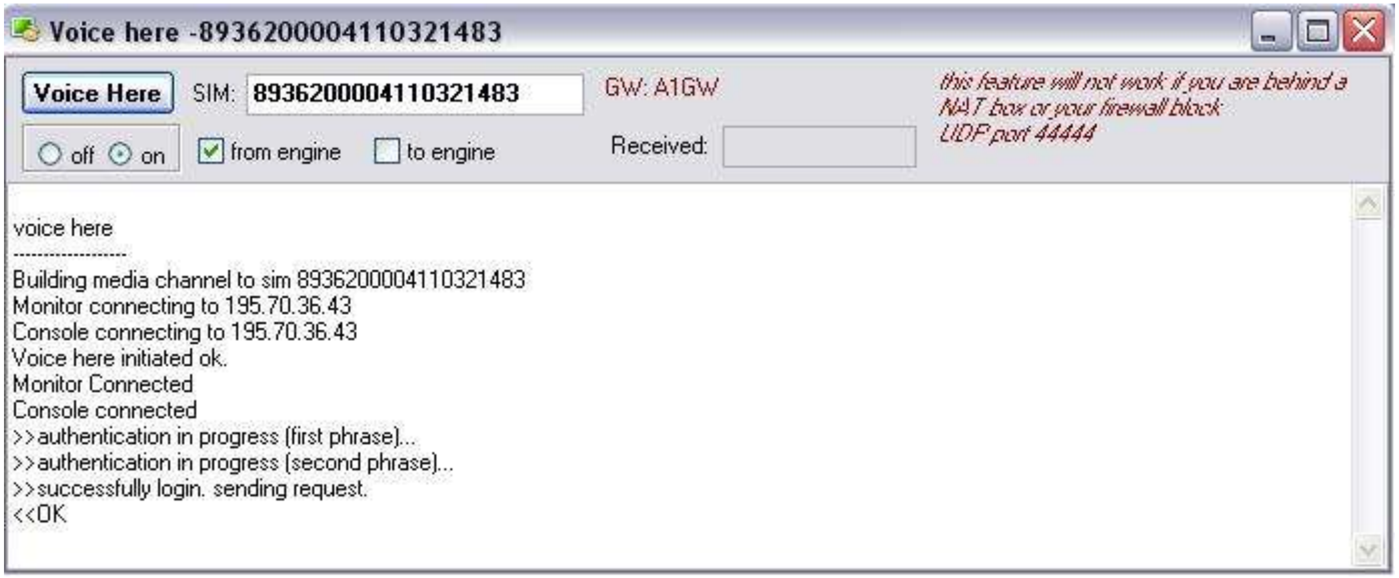
> 0=don't show,1=show only spouose, 2=only child, 3 = show both of them.
> default value is: 3

4.8.2. Direct Query

From this form, direct SQL queries can be done against the Mizu backend. Use it carefully!

4.8.3. Voice Here

With this utility, the conversations on Mizu gateways can be listened in real-time.



Voice here -89362000041103214 83

SIM: 89362000041103214 83

GW: A1GW

○ off ● on ☑ from engine ☐ to engine

Received:

this feature will not work if you are behind a NAT box or your firewall block UDP port 44444

```
voice here
------------------
Building media channel to sim 89362000041103214 83
Monitor connecting to 195.70.36.43
Console connecting to 195.70.36.43
Voice here initiated ok.
Monitor Connected
Console connected
>>authentication in progress (first phrase)...
>>authentication in progress (second phrase)...
>>successfully login. sending request.
<<OK
```

4.8.4. Test Call

VoIP test calls can be done here.

4.8.5. Rfile system

Upload/download files from gateways.

4.8.6. Rdesktop

Use this form to login directly in gateways and on the server.

4.8.7. DB Admin

Database administration tool. Only for database experts!

4.8.8. Web Admin

Direct link to the Costumers website if you have any.

4.8.9. Phone Numbers

Numbers allocated by authorities. You may add new endusers with telnumbers set to a free number from this database. Don't forgot to set the "free" field to 0 if the number is allocated to an enduser.
The web interface will get free numbers for newly registered users from this database too.

4.8.10. To-do

You can define tasks for technical support with the ease of this form.

4.8.11. Notes

Any quick note here to be shared between the support and admin users.

4.8.12. Holidays

Mizu server can treat holidays as weekend in *routing* and *billing* if the "**autoholiday**" global config is set to "1".
If the autoholiday is set to 0, than you can configure the holiday routing and billing manually (by selecting the "holiday" pattern from the time pattern list)
In the **"Holidays" Form** you can set the holidays.
The *isholiday* value can have the following values:

1: the time period is considered as holiday
0: the time period is considered as workday (even if is weekend)

Then you must set the time interval (*fromdate-todate*).
Prefixlist can contain '*' or 2,3 or 4 digit prefixes separated by comma.

If the prefix list is empty or it contains '*' than will be applied to for all directions.

4.8.13. Allocating numbers

Allocated and used phone numbers can be tracked on the "Phone Numbers" form.
Numbers allocated to users or sipproxyes must be marked (set "**free**" to 0)
The **location** can contain the id of the user where the number belongs. This is very important for sip proxy users where the number is not stored as the record username or number (usually for virtual servers). If set so, the server will know where to **route** the incoming call. Otherwise the routing must be configured in the MManage.

4.8.14. Scheduled tasks

With the scheduled tasks form you can define tasks that will be executed by the server at regular time intervals. Two type of task is allowed: launching as program or running an SQL command.
Depending on completition the server can initiate the following actions:

- Send Email
- Send SMS
- Ring a phone number
- Restart the VoIP service
- Restart the SQL service
- Reboot the server

For SQL the condition will be success if the first field value is a positive number (first row/first col). Otherwise (no record, null record, non number, zero or negative number) the result will be treated as "fail".

The following keyword can be used with the email and sms actions:

- SQLRESULT: will be replace with the sql query result set (all rows and columns)
- {FIELD}: any field returned by the query (replace the FIELD with a valid returned field from your query)
- {SQL}: any SQL result (replace the SQL with a select statement like {SELECT TOP 10 * from tb_cdrs})
- USERID: it can be used in the above SQL to be replaced with tb_users.id if the query was against tb_users
- ***SUBJECT: anysubject (to specify a subject for the message
- ***TO: email@domain.com (to specify an exact address for the message to send to. Otherwise it is sent to admins and support)
- ***TO: USER: this is a special keyword if your query is against the tb_users table so all record will be processed separately and the message will be sent to current user email or phone.

For example to send daily notifications to users with credit below $10, you might set the followings (you need to create a lastemailnotificationsent field in tb_users):

- Interval: daily
- Type: conditional

- SQL Query: select id,email,name from tb_users with(nolock) where credit < 10 and postpaid < 1 and enabled <> 0 and temporarydisabled <> 1 and lastemailnotificationsent < getdate() - 10
- Action On Succ: email
- Action On Fail: no action
- Message body:
  ***SUBJECT: Low Credit on VoIP
  ***TO: USER
  Dear {name}
  Your credit is {credit}
  {update tb_users set lastemailnotificationsent = getdate() where id = USERID}

Example to use SQL keyword (sending the last called number for a users every night):

- Message body:
  ***SUBJECT: Your last call
  ***TO: USER
  Dear {name}
  Your last call was made to {select top 1 callednumber from tb_cdrs with(nolock) where datum > getdate()-1 and callerid = USERID order by datum desc}

4.9. Call Center

This section will describe the outgoing callcenter module but can also be useful to read when using the incoming callcenter module (e.g. IVR)
The Mizu callcenter combine the maximum efficiency with easy to use, intuitive interfaces.
Statistics can be viewed in real-time. See Callcenter Statistics for more details.

4.9.1. Users

Will load the callcenter operators (agents). Here you can add, delete and edit them.
The basic settings are placed on the **Edit Operator** tab. SIP enduser related settings can be edited on the **Advanced** tab.
With the Campaign drop-down list you can assign the selected operator to a campaign.
Operators must have entered and quit date set correctly. (If the quit date is elapsed, than the operator is not allowed to work with MAgent)

Technically operators are just sip endusers (tb_users.type =0) by the isoperator flag is set to 1.

4.9.2. Campaigns

You can setup the campaigns in this form.
Campaign will start to run when the StartDate is reached and will run until no more clients (phone number) are assigned or the EndDate was reached. This means that MAgent -> Automatic Calls will run if this conditions are met.
The "Display" field will be displayed for the operators in MAgent "Automatic Call" window.
By clicking on the "Load Statistics", a sort statistics window will be displayed regarding the selected campaign.

Handling invitations:
Load Invitation: will download the assigned invitation from the database. This can be any file, but Microsoft Word document are preferred.
Save Invitation: will save the document back to database. Prior to hit this button, the document must be edited, saved and closed.
Print Invitations: will print a separate invitation for all invited clients in this campaign.
You can use special keywords in word documents and that will be replaced with the corresponding value. See the Keywords section for more details.

4.9.3. Scripts

Every campaign can have different operator instructions. These instructions can be defined in this form.
For every step (question) the operator can select from different actions (answers). The call will follow these selected instructions. Pay attention to cover all possibilities.

tb_ccscripts: (used for questions)
id: autoincrement PK
campaignid: the campaign where this script belongs to
ordernum: display order (used when jumping to next question)
title: script alias. also used for statistics
question: question in text format (can contain keywords. see the Keywords chapter for more details)
questionex: the question in rtf or html format
completition: completition marks can be used for statistical purposes.
oldid: used for cloning
datainputtype: used to control the gui for the anwer

-----SIMPLE-----
Text
Number
Date
Time
Date-Time
Phone Number
CheckBox
-----LISTS-----
CheckBoxList
ListBox

RadioGroupList
DropDownList
DropDownListFixed
-----GRIDS-----
RadioGrid
CheckGrid
-----COMPLEX-----
Custom GUI
Run
-----ACTIION-----
ActionRecall
ActionReject
ActionAccept

<u>tb_ccscript_answers:</u> (used for answers)
id: autoincrement PK
questionid: answer belongs to this question
ordernum: display order
answer: the answer text
alias: will be stored as answer (usually the same as the 'answer')
actioncontext: can be used for storing reject reasons or for data imput field specification
fields: used grids to determine its columns. Values must be separated by comma ',' or semicolon ';'
jumpto: next presented question. possible values:
-3: no jump
-2: jumt to next question
-1: finished
-4: finised with voice
-5: jump forward
other: questionid
endcode: the endcode for the last answer (with a valid code) will be stored for statistical reasons. if you specify more endcodes, the last endcode will be stored in the database.
<u>entercondition</u>: simlple sql clause. if the condition result is true, than will jump to question 'onconditiontrue', othervise will jumpt to 'onconditionfalse'
entercondition rules:
1. any sql in the following format: select … (if return 0 than the result will be interpreted as false, othervise as true)
2. simplified condition. can use the following keywords:
prefixes:
any database table name (tb_cclient, tb_ccampain_clients, etc)
script (maps to scriptquestion.alias - scriptanswer.alias)
client (maps to tb_cclient)

cclient (maps to tb_ccampain_clients)
campaign (maps to tb_cccampaigns)
quotacount, quotapercent, completedcount, completedpercent, averagequotacompletition

suffixes: script alias names or table fields name

operators and functions: most of sql92 keywords will work (=, <>, <, (,),TRIM, LOWER, AND, OR, etc)

examples:

1. (script.alias = 'fidesz' or scriptcode.alias = 3 or quotacount.quotaalias >0) and (LOWER(client.name) = 'john')

2. quotacompletedpercent.quotaalias < 100

see the keywords section for more details!

onconditiontrue: if the entercondition result is true, than the next question will be the question with this order (ordernum)

onconditionfalse: if the entercondition result is false, than the next question will be the question with this order (ordernum)

Running external applications:

set the datainputtype to "Run"
specify application name and parameters in the "Command" field.
keywords can be used as part of command.
the following Parameters are defined (separe them with comma)
-hide: will launch the application hided
-wait: will wait for the application to terminate

Text Mask:

Use EditMask to restrict the characters a user can enter into the masked edit control to valid characters and formats. If the user attempts to enter an invalid character, the edit control does not accept the character. Validation is performed on a character-by-character basis by the ValidateEdit method.

A mask consists of three fields with semicolons separating the fields. The first part of the mask is the mask itself. The second part is the character that determines whether the literal characters of a mask are saved as part of the data. The third part of the mask is the character used to represent unentered characters in the mask.

These are the special characters used in the first field of the mask:

! If a ! character appears in the mask, optional characters are represented in the EditText as leading blanks. If a ! character is not present, optional characters are represented in the EditText as trailing blanks.

> If a > character appears in the mask, all characters that follow are in uppercase until the end of the mask or until a < character is encountered.

< If a < character appears in the mask, all characters that follow are in lowercase until the end of the mask or until a > character is encountered.

<> If these two characters appear together in a mask, no case checking is done and the data is formatted with the case the user uses to enter the data.

\ The character that follows a \ character is a literal character. Use this character to use any of the mask special characters as a literal in the data.

L The L character requires an alphabetic character only in this position. For the US, this is A-Z, a-z.

l The l character permits only an alphabetic character in this position, but doesn't require it.

A The A character requires an alphanumeric character only in this position. For the US, this is A-Z, a-z, 0-9.
a The a character permits an alphanumeric character in this position, but doesn't require it.
C The C character requires an arbitrary character in this position.
c The c character permits an arbitrary character in this position, but doesn't require it.
0 The 0 character requires a numeric character only in this position.
9 The 9 character permits a numeric character in this position, but doesn't require it.
# The # character permits a numeric character or a plus or minus sign in this position, but doesn't require it.
: The : character is used to separate hours, minutes, and seconds in times. If the character that separates hours, minutes, and seconds is different in the regional settings of the Control Panel utility on your computer system, that character is used instead.
/ The / character is used to separate months, days, and years in dates. If the character that separates months, days, and years is different in the regional settings of the Control Panel utility on your computer system, that character is used instead.
; The ; character is used to separate the three fields of the mask.
_ The _ character automatically inserts spaces into the text. When the user enters characters in the field, the cursor skips the _ character.

Any character that does not appear in the preceding table can appear in the first part of the mask as a literal character. Literal characters must be matched exactly in the edit control. They are inserted automatically, and the cursor skips over them during editing. The special mask characters can also appear as literal characters if preceded by a backslash character (\).

The second field of the mask is a single character that indicates whether literal characters from the mask should be included as part of the text for the edit control. For example, the mask for a telephone number with area code could be the following string:

(000)_000-0000;0;*

The 0 in the second field indicates that the Text property for the edit control would consist of the 10 digits that were entered, rather than the 14 characters that make up the telephone number as it appears in the edit control.

A 0 in the second field indicates that literals should not be included, any other character indicates that they should be included. The character that indicates whether literals should be included can be changed in the EditMask property editor, or programmatically by changing the MaskNoSave typed constant.

The third field of the mask is the character that appears in the edit control for blanks (characters that have not been entered). By default, this is the same as the character that stands for literal spaces. The two characters appear the same in an edit window. However, when a user edits the text in a masked edit control, the cursor selects each blank character in turn, and skips over the space character.

Setting EditMask to an empty string removes the mask.

tb_ccscript_processing: (used for storing answers)
id: autoincrement PK

datum: row age
operatorid: the id of the caller operator
clientid: the id of the called client (from tb_cclient)
ccid: campaign_client foreign key
questionid: the id from tb_ccscripts
answerid: the id from tb_ccscript_answers
answervalue: entered value (saved as text, but can represent other data type. for example date-time)

Note:

data inputs can be saved to tb_cclient or tb_campaign_clients if specified so. column names must be prefixes with client. or campaign. keywords. see the keywords section for more details

all script answers (including data input) will be saved to tb_ccscript_processing
list answers will be separated by semicolon ;
grid answer will be saved in separate columns where the column title will be the tb_script.title + the grid column title

4.9.4. GUI Designer

The MAgent Automatic Call form user interface can be customized in the MManage -> GUI Designer.
Basic MSWindows controls and specific controls are supported.

*Note: the dynamic MAgent gui will be loaded only if the "***autoformname***" setting is set to "dynamic". Otherwise other company specific form will be loaded.*

Typical workflow:
1. Load a default gui from file
2. Drag and drop controls on the gui area
3. Set controls properties
4. Save the new gui in the database (the "New" checkbox must be checked, otherwise will overwrite!)
5. Set campaign gui to the newly created one (or set it to a script question as a data input gui)

Text controls can contain **dynamic data** if the control "Data" properties are defined in the following way:
-[tablename.fieldname]: simple map to the required field
if the control is not read-only, than the edited value will be saved back in the database
-[select …]: any select condition
-only the first column in the first row will be considered
-keywords (see the Keywords chapter)

Specific controls:
-Script question: space for displaying script questions
-Script answers: answer list. Deprecated!
-Script data: space for displaying script answers and choices
-Recall label: will display the call “recall” status
-ProgressBar: for displaying the call progress (the position in the script)
-Ok: save script answer and jump to next question
-Skip: jump to next question without saving the current
-Hangup: dropp the current call
-Pause: no more new calls
-Next Client: force jump to next client (will hang-up the current call if any)
-Recall: set client for later recall
-Reject: will set the rejected flag
-Hold: call hold (mute all directions)
-Display call status: display called party name and call duration
-Call Statistics: operator statistics

4.9.5. Quotas

You can define campaign target quotas by launching the “CC Quotas” form from MManage.
By defining quotas, you can restrict your calls to well defined target groups (called clients).

Database fields:
id: autoincrement PK
~~allowed: deprecated~~
campaignid: foreign key to tb_cccampaigns.id
alias: the name of the quota (any string here)
condition: sql clause applied for tb_cclient and tb_ccampain_clients. All separate expressions must be included in barckets!
found: found clients after fixed condition
quotacount: number of desired interviews
quotapercent: percent of desired interviews from global quota. the global quota can be specified in the campaign settings, or is calculated automatically
status: any number. you can refer to it from scripts. for example when you don’t want more calls for a quota, set the status to 1
and check it in the script enter condition (if quotastatus.myalias = 1 than jump to finish)
~~autobalace: calls will be automatically balanced based on condition (if the condition can be evaluated)~~
~~otherwise the quota completition percent doesn’t effect the calls~~ deprecated
completedcount: number of completed interviews

completedpercent: competed interviews in percent
enabled:

-0: ignored (instead of deletion)
-1: ignored on server (but can be referenced from scripts)
-2: default

Block: if set to 1, than calls to clients that meets the quota condition will be disallowed
Autostop: if set to 1, than call will be blocked when the completedpercent reach 100%

Either quotacount or quotapercent need to be specified.
If the global quota for the campaign is not specified, and only the quotapercent is specified for the campaign, than the quota will remain invalid.

Note: call record must reach a completition question to be valid in quota calculations

4.9.6. Presentations

Used to store the different presentation locations. When a client is invited, the operator will select a presentation for them.
Presentation can be selected dynamically in MAgent by setting the sql condition in the script form.
It can be done by tree way:

1. leaving empty
   In this case the following query will be run: **select top 150 id,Name,startdate,remainusers,comment from tb_ccpresentations with(nolock) where > getdate() and campaignid = :assignedcampaign order by startdate, name**
2. using only the sql clause
   for example: **name like '%hotel%'**
   the default clause is: **startdate > getdate() and campaignid = :assignedcampaign**
3. using a complete sql.
   In this case the "id" and a "displaytext" field must be returned.
   Example: **select id, name+' '+comment as displaytext from tb_ccpresentations where name like '%hotel%'**

The usual keywords can be embedded in the sql scripts. See chapter Keywords.

4.9.7. Checklist

Can be used in presentations to print the list of invited users.
Status texts can be configured in tb_ccstatus. To be compatible with other version, change the texts to correspond with the original without changing the statusid.
The "Print Invitations" button will print all invitations in the selected campaign/checklist whose status indicated that it was not printed prior.

4.9.8. Clients

The client (phone number) database.

Clients can be assigned and/or reassigned to campaigns with the ease of this form.
You can search across client by a lot of condition presented on this form.

By selecting the “Last Status” filer, the users can be searched by the reason code in the last campaign
By selecting the “Any Status” filer, the users can be searched by the reason code in any campaign

Importing client database can be done from external csv or dbf files. These files must have the following fields:
CSV file columns (must be in this order):

- -Name (string)
- -Landline phone number (string)
- -Mobile phone nuber (string)
- -Zipcode (short string)
- -City (string)
- -Age (number)
- -Passport (0 if unknown, 1 if no or 2 if yes)
- -Married (0 if unknown, 1 if no or 2 if yes)
- -Sex (0 if unknown, 1 if no or 2 if yes)
- -Robinson (0 if unknown, 1 if no or 2 if yes)
- -Address (string)
- -Comment (string)

DBF files must contain the following columns (can contain other columns too):

- -UNEV: user name
- -TEL: phone number
- -VAROS: city
- -UTCA: address
- -ROBIN: robin
- -IRSZ: zipcode

At least the landline or mobile phone must contaion a valid entry.

4.9.9. Campaign Clients

For every campaign clients must be assigned. This means loading records from tb_cclient to tb_ccampain_clients. Records from tb_cclient are usually not modified during the campaign. All modification will be done in tb_ccampain_clients and in tb_ccscript_processing (scrip answers).

<u>tb_ccscript_processing database fields:</u>
id: PK, unique identifier (autoincrement)
campaignid: foreign key pointing to tb_cccampaigns.id
clientid: foreign key pointing to tb_cclient.id
clientname: first characters from tb_cclient.name. (denormalized here for speedup)
datum: client record assigment date

operatorid: last operator who have called this client
aquired: used in automatic calls for record locking. deprecated.
enabled: if 0, than this client will not be called again
calltrycount: number of call attempts
recalltrycount: number of recall attempts
lastfailtype: wich number has failed last time (0=unknown,1=mobile,2=landline)
called: the client has been called successfully
needrecall: 0: no need for recall, 1=recall set manually by the operator, 2=recall automatically due to unhandled incoming call,3=other recall reson, >=4: already recalled
recalldate: date-time of the next recall attempt
lastcalldate: date-time of the last call attempt (successfully or failed call)
comment: any comment (can be filled by operators)
randorder: deprecated
numstatus: wich number is present. 0=both are missing,1=only landline,2=only mobile,3=both (denormalized here for speedup)
guidrand: used when clients need to be called randomly (not in database or A-Z order)
rejected: 1 if the call was rejected (rejected by user not by the phone)
rejectedreason: optional comment when rejected

4.9.10. Campaign and global settings

Allowdbcalls: allow calls from database in MAgent
Allowmanualcalls: allow manual calls in MAgent
Callmaxring: max ringtime when automatic calls in sec
Callnumbertype: 0=start with landline, and if fail, call mobile, 1=start with mobile, and if fail, call landline, 2 =call only landline, 3 = call only mobile
Maxcalltrycount: max number of calls to a client (except recalls).
Maxrecalltrycount: max number of recalls to a client.
Recallrestrictions: 0=try to recall with the same operator, but allow other if no recall, 1 = only with the same operator, 2=any operator can recall, 3 =disable recalls
Predectivedial: predective dialing: 0=don't use, 1=onfree, 2=onfree + periodic,3=periodic,4=automatic,>4=automatic will switch to 2 if this number is operator reached
predectivelogging: will save detailed logs about predective activities and statuses
waitforpredective: timeout in sec to wait for a free operator on connect. 0 disables waiting. -1 will disconnect all calls in progress (ringing) when there are no more free operators.
Predectivecorrection: initiated call count correction in percent (for example 80 will generate calls faster by 20%)
Callbackhandling: 0=dropp calls, 1=forward to the same UE if exists, 2=forward to a free operator, *3=try to forward to a operator but on fail route to the same UE, 4= forward to UE and if not exists than to a free operator , other=forward to the specified number
Callbacknumber: special threatments when this number is called. When the predective caller user is already created, than the callbacknumber is its username

Callbackroutenumber: called operator when callbackhandling is 1,3 or 4. defaults to callbacknumber
ccorder: determine how clients are loaded

0: campaing client id
1: load calls from A to Z
2: load calls from Z to A
3: load calls in random order
Other: not ordered

waitifnotconnected: wait after calls (callmaxwait sec) even if it is not connected (to allow opertor work)
callbackringtimeout: ring timeout on callback (after than play ivr message if set)
callmaxwait: max waittime allowed for operators between calls (for administrative purposes)
callbackivr: play special messgae (callback.wav) if no operator found or ring timeout expired
callbackhandling: 0=dropp calls, 1=forward to the same UA if exists, 2=forward to a free operator, 3=try to forward to a operator but on fail route to the same UA, 4= forward to UA and if not exists than to a free operator , other=forward to the specified number
callbackautorecall: 1= schedule for recall if number is in campaign,0=don't recall automatically
defrecallmin: the callback date-time selector will appear with this default recall minue
maxrecallmin: max minute setting allowed for callbacks in MAgent
finishvoice: calls will be finished with this voice promt
mobileratio_X_Y: the percent of mobile calls between hours X and Y
presentationmode: if to show presentation choice in MAgent
recallonlyinsamecampaign: recalls allowed from other campaigns too
desireddropprate: optimal percent of calls wich cannot be assigned to operators (-1: disable, -2: dynamic)
maxcallsperminute: max new call attempts/minute (-1: disable, -2: dynamic)
maxcallatonce: max call launch/branch (used for additonal checking)
statival: statistics recalculation interval in sec (-2: dynamic)
stopwrongcdrc: will pause predective execution for 15 minute if subsecvent “stopwrongcdrc” calls are less than 8 sec to guard agains service provider error
scriptnavigation: allow script navigation in MAgent (0=no, 1=only forward, 2=only back, 3 = forward&back, 4 =all navigation, 5=allow navigation when not connected, 6=allow edit too)
datainputmode: 1=store only in client and cc_client tables, 2=store in cc tables and to default field to, 3=store in cc tables and in script processing, 4=store in all locations
savemode:

1=automatic –will save on when moving to next question (don’t display ok button)
2=save only when the ok button was pressed
3=with skip button
4=ok+automatic
5=cannot skip. user must press ok button to be able to move to next question

blockrejected: when the tb_ccampain_clients.rejected is set to 1 by operators, than will set tb_cclient.robinson to 1 automatically
loginmessage: message to be displayed for the operators on MAgent launch

allowopcampchange: allow operators to change their campaign
maxrecallbefore: callbacks can be initiatet "maxrecallbefore" minute before the specified time
maxrecallafter: callbacks can be initiated by the operator untill "maxrecallafter" minute after the specified time. if the operator is not able to receive the call, than the call will be offered to other operators after maxrecallafter elapsed (and untill maxrecallafterall)
maxrecallafterall: callbacks will be invalidated if not able to give to any operators when this timeout elapse
quotastatrecalcival: quota statistics recalculation intervall (specified in second)

Note: the predictive dialer is not so precise when started after a long time of inactivity. Accuracy also depends on the number of operators (more operator means more accuracy).

4.9.11. Predictive dialer

To restrict the operator wait times, the calls can be prepared on the server side and dropped to operators when they are waiting for it.
There are separate predictive threads for all campaigns. The campaign predictive related settings will overwrite the global configuration
The predictive thread will check if new calls are needed in every 1-20 sec (varies dynamically)
Calls are started based on the following variables:
-predictive mode
-1: new call will be made only when waiting operators found
-3: new calls will be started regularly based on predictive calculations
-2: mix of 1 and 3
-4: will switch between 1 and 2 dynamically
-operator count (all, active, waiting)
-call statistics (call count, asr, acd, dropprate) calculated on long term, midterm and short intervals
-campaign and global configurations (desireddropprade, etc). Please check Global Settings for more details.
The thread activity can be logged in tb_predectivelogs, otherwise the standard logs can be found when searching to "predictive".

Calls are started by the PREDECTIVE_DIALER user (this will be created automatically if not exists), so the A number will be its number (loaded from the „callbacknumber" config when created automatically). When the called party pickup the phone, a free operator will be searched and the call will be towarded to it. So the call is already connected when it is presented to the operator. If the operator cannot handle the call, than a CDR record will be generated (usually with a short duration) but with a specified reason code. Another special case is when there is no free operator found when the call is connected. In this case the disconnect reason will be set to „No free operator on predective".
If the embedded dialer speed dosn't satisfy your need, you can control the speed by changing the desireddropprate and the predectivecorrection configuration. The predective functionality perform better as there are more operators (more than 10).

Predective related settings:

- Callbacknumber: "A" number for calls. For example the predective dialer will use this number
- Desireddropprate: optimal percent of calls wich cannot be assigned to operators when in predective (-100 has no effect)

- Maxcallatonce: max number of calls in one round when in predective (error guard)
- Maxcallsperminute: max new call attempts/minute when in predective (error guard)
- Predectivecheckival: controlls the speed of the predective dialer thread -advanced technical setting
- Predectivecorrection: correction of precalculated success ratio statistics in predective. for example if we set it to 80, than there will be more calls with 20% than the precalculated optimal
- Predectivedial: dialing mode: 0=simple MAgent requests,1=automatic on free clients,2=predective inteligent, 3=simple predective, 4=set automatically
- Predectivelogging: details of predective logs (0=no logs,1=minimal,2=nnormal,3=max log)
- Statival: rebuild predective statistics interwal (-2 = automatic)
- Stopwrongcdrc: pause predective if the ASR is too low (error guard)
- Waitforpredective: max time to wait for a free operator when a predective call is connected
- Waituntillconnected: if set to false, than will dropp started call when no more operator waiting (no more "No free operator on predective" disconnect reason)

4.9.12. Outgoing callback

There are two types of callbacks:
1. operator set **Recall** (1) manually
2. server receive an incoming call and set to **Callback** (2) automatically

Client records will be marked when callbacks or recalls are needed. The following values are defined for the „needrecall" field:
-0: not set
-1: recall set by operator
-2: callback set because incoming call
-4: recalled
-5: callback completed

Callback numbers:
Special A numbers are the followings:

- callbacknumber
- predective dialer number
- identityrewrite number for outgoing proxy

these should be set to the same number. By default it is loaded from „callbacknumber" global setting
**if the callbacknumber is not set, the predictive dialer will use "1111" by default*
** identityrwmode global setting should be set to 1 for callcenters*
**if the CLI is hided, than there will be NO incoming calls,*

to enable CLIP you must set the "identityrwmode" setting to 1 or set the CLI for users to 1

The „callbackhandling" global setting will determine how incoming calls to callbacknumber will be handled. Incoming calls when the caller is not found in any campaign will be inserted to the client table with a name set to "unknown callback".

Make sure to setup the callback number on the main server to be routed to the actual virtual server! (check faq)

4.9.13. Incoming calls

Incoming calls can be handled in different ways:

1. Simple distribution across operators
2. By specifying a callbacknumber and setting the callbackhandling option.
3. Handling by the IVR
4. Make incoming campaigns

Incoming callers (clients) identity can be loaded from database when the call arrives. If the client data is not found, then can be added to database automatically.

In the MAgent the call can be handled in the following ways:

1. Dropp all
2. Show in Manual Call
3. Show with client data
4. Show in "Auto Calls" with the actual script

The following config settings will be applied:

CampType:

0 or NULL=default
1=callcenter (OUT)
2=IVR (IN) -new numbers will not be requested from the server
3=Mixed -incoming calls can be received from ivr, but from dialer too

Callbackhandling:

0=dropp all
1=route to callbackroutenumber
2=route to free operators
3=route to free operators, and if not answered than route to callbackroutenumber
4=first to callbackroutenumber than to free operators

Callbacknumber:

"A" number for calls. For example the predective dialer will use this number

Callbackringtimeout:

ring timeout on callback (after than play ivr message if set)

defrecallmin ring timeout on callback (after than play ivr message if set) defrecallmin

callbackroutenumber:

number to be dialed on incoming calls when callbackhandling is 1,3 or 4

ivr_addmissingusers: -applicable on serverside ivr calls and in MAgent

0=no -default

1=add to client table

2=add to campaign client

addmissingusersto: -in wich campaign to add the incoming client

0=add to the current campaign (default)

other=add to the specified campaign (campaign id)

Handlingincoming:

0=not handled

1=allow to search the current campaingn for the incomig number

2=allow to search and edit the current campaingn for the incomig number

3=allow to search the whole database for the incomig number

4=allow to search and edit the whole database for the incomig number

5=show scripts form

~~*Handlingincoming:*~~

~~*0=not handled*~~

~~*1=allowed*~~

~~*2=show scripts form*~~

~~*IncSearch*~~

~~*0=no automatic search*~~

~~*1=allow to search and edit the current campaingn for the incomig number*~~

~~*2=allow to search the whole database for the incomig number*~~

~~*3=allow to search and edit the whole database for the incomig number*~~

Incoming campaigns:

1. First you must define your phone numbers wich will handle the incoming campaign(s).
2. If the server is a virtual server, you must setup your main server to route that numbers to the virtserver (See this caption for more details)
3. You must add this numbers to your user list, and assign an IVR for them.
4. For the IVR campaign set the “CampType” field to 2 and don’t forget the ivr_addmissingusers and handlingincoming settings
5. Create the MAgent script and GUI in the MManage.
6. Assign some operators for the campaign(s).

4.9.14. Keywords

The following controls can contain keywords:

-script question
-script answer
-script enter condition
-quota condition

Table name abverbiations:

camp => tb_cccampaigns
client => tb_cclient
campaign => tb_ccampain_clients
script => tb_ccscripts with answertext
scriptcode => tb_ccscripts with code
scriptquestions => tb_ccscripts
scriptanswers => tb_ccscript_answers
quota(s) => tb_ccquotas

The following <u>keywords</u> are defined:

-all **database fields** from tb_cclient, ccampain, tb_ccampain_clients, tb_ccscripts, tb_ccscript_answers and tb_ccquotas
-[currentnumber] current called number
-[callduration] call duration
-[ringduration] ring duration
-[callstarttime] begin time
-[callcount] number of call attemts
-[currdatetime], [currdate], [currname] current time/date/date-time
-[currweekday], [curryear] weekday, year, month, min, etc
-[bell] bell
-[campaignname] campaign name
-[opusername] operator username
-[opname] operator name
-quotacount
-quotastatus
-quotapercent
-quotacompletedcount, completedcount
-quotacompletedpercent, completedpercent

All ep specific keyword list:

currentnumber
lastforwardednumber
callduration
ringduration

creditdurationmin
creditdurationminminpx
creditdurationminmin
creditduration
creditdurationmin2
creditduration2
maxspeachlen
credit
credit_tospeech_ex
credit_tospeech
bell
companyname
opusername
dtmf
dtmftrimmed
dtmfstored
storedivrstring
storedani
dtmfstoredtrimmed
opname
currdatetime
currdate
currdatetimesql
currtime
currweekday
curryear
currmonth
currday
currhour
currmin
currsec
calleruserid
calleduserid
proxyid
origcallernumber
origcallernumber_nq
orignumber_nq
realcallernumber
callernumber
callernumber_nq
number_nq
callername
origcallednumber
callednumber
calledname
techprefix
called_norm
auth_username
auth_password
transportip
fromip

rtprecip
transportport
fromport
rtprecport
callertype
usertype
callstate
cc_clientid
cc_ccid
credit
maxduration
globalcurrency
ratingcents
ratingcurrency
rating
rating_tospeech_ex
rating_tospeech
currency
centname
disccausetextex
otherpartyname
otherpartydisplayname
otherpartyfullname

Embedding controls in texts:

Basic:

Keyword embedded in square brackets [xxx] will be replaced with dynamically loaded data. (Label)
Keyword embedded in brackets {xxx} will be editable (loaded and saved to/from edit control) (Edit)

Advanced:

You can define other controls by the following text rule:
{ControlName|Width|BindValue|Item1|Item2,...}
The following control types are defined: Edit,Memo,ComboBox,ComboBoxRo,CheckBox,RadioButton,Date,Time

Examples:

```
{ComboBox|150| tb_ccampain_clients.fawoYiteos|Mac|Windows|Linux}
{ComboBox|200|tb_cclient.phone_landline|elsooo|masodiiik|harmadiik}
{Date|300|tb_cclient.recalldate}
```

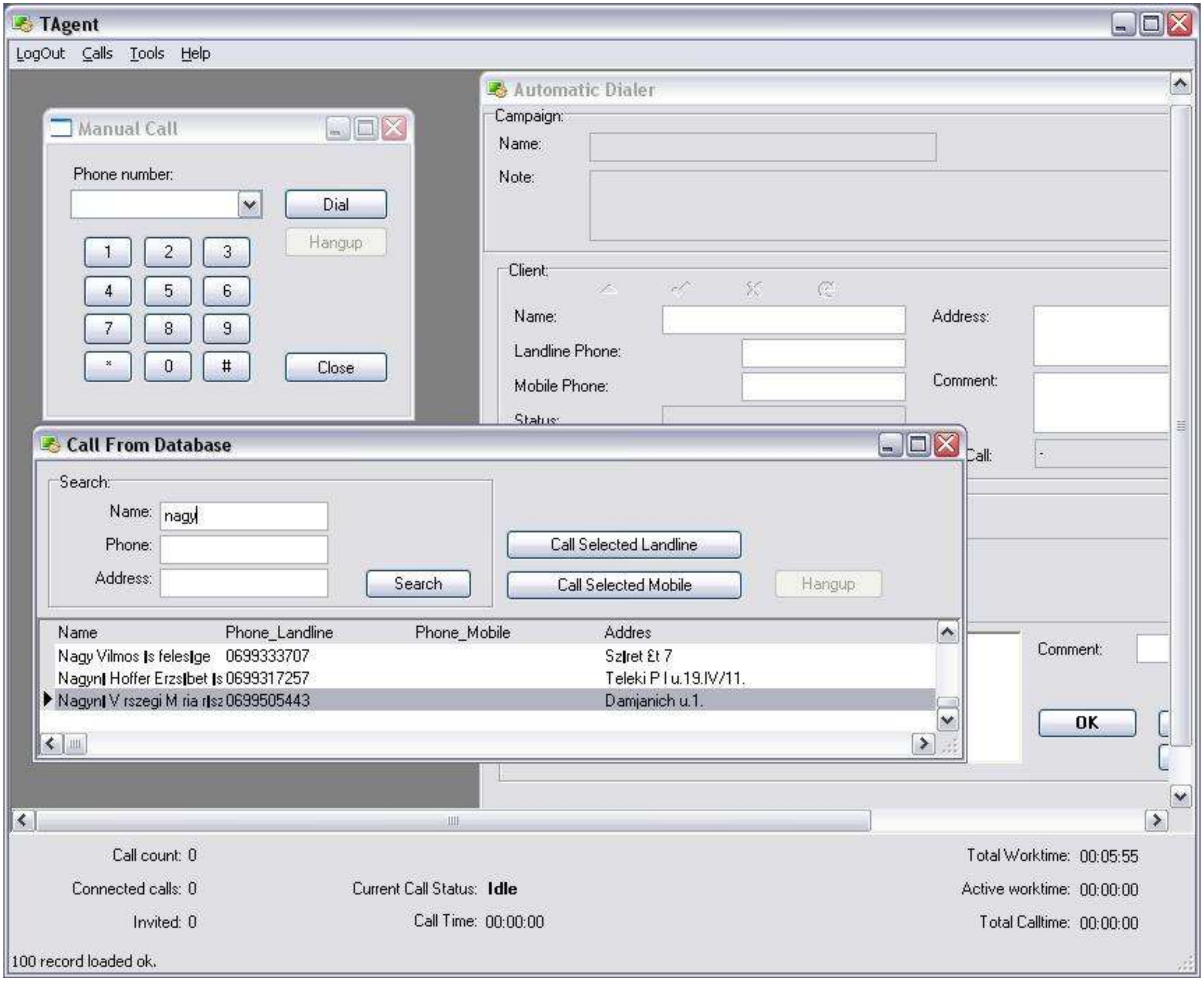
Controls that can contain keywords, can also contain simple select conditions.
For example for list controls in the answer field a simple sql query can be specified wich can return one or two columns, wich will represent the list items.
If two columns are returned, the first should be a number (key).

4.10. MAgent

The MAgent application is used by callcenter operators as a sip client and database frontend.

4.10.1. Login

Enter server settings and authentication info here to login.
The following values are required on login:
App Server: server ip address
Instance: Application and database instance (because a single server can hold several virtual server)
Data port: defaults to 1433 (or 2223 for older versions)
Database username: the same for all agents
Database password: the same for all agents
Username: agent username
Password: agent password



Name: nagy

Name	Phone_Landline	Phone_Mobile	Addres
Nagy Vilmos ▌s feles▌ge	0699333707		Sz▌ret £t 7
Nagyn▌ Hoffer Erzs▌bet ▌s	0699317257		Teleki P l u.19.IV/11.
▶ Nagyn▌ V rszegi M ria r▌sz	0699505443		Damjanich u.1.

Call count: 0
Connected calls: 0
Invited: 0

Current Call Status: **Idle**
Call Time: 00:00:00

Total Worktime: 00:05:55
Active worktime: 00:00:00
Total Calltime: 00:00:00

100 record loaded ok.

4.10.2. Manual Call

Simple VoIP client window where the operator are free to make calls to any number

4.10.3. Calls from database

Call to any client presented in the central database.

4.10.4. Automatic calls

Will handle calls automatically if the operator is part of a campaign.

4.10.5. Automatic software upgrades

Mizu client applications can check for newer version and **selfupgrade**.

The following modules are responsible for handling software updates:
-TUpdater.exe: to download the new software from a central directory (usually ftp)
-TUpdLaunch.exe: will check for new downloaded software on every main software start (will make backup of the old sw)

For autoupgrades to work the following tasks must be done:
-copy the new software's in the **ftp**/specifieddirectory
-insert a new record in **tb_autoupdate**
swname: the name of the main software executable that needs to be updated (for example: MAgent)
mode: 0: ask the user first, 1: automatic install
version: new software version. make sure that this is correct especially when you use * in "oldversions" to avoid endless update!
oldversions: client softwares will check this record to know if the new software can be applied (* mean all)
forusers: applicable for this users (* or database user names separated by comma –*login names from tb_users*)
host,username,password,dir: ftp parameters. make sure than the user will have access to the files!
files: files that must be updated (separated by comma)
comment: will be displayed for users (new features, corrected bugs, etc)

Make sure that the version of the new software is bigger than the current and is not in the "oldversions" list, because otherwise will result in an endless update proces!

4.11. PBX / IPCentrex functionality

In addition to supporting class5 features according to sip rfc's, drafts and recommendations, most of the features normally held on UA are implemented on the server side also and can be accessed by dtmf messages (client device must be able to send DTMF as INFO or RFC2833 at least)

4.11.1. Call Rerouting

Failed calls (error code, no answer) will be rerouted automatically to the next device according to routing preferences.
The rerouting can be performed within session, so the caller user doesn’t notify that the call was rerouted.
This is different from call failover (which means that a route is marked as low priority temporarily or definitely).

4.11.2. Ring Groups and Call Fork

A call can be sent to multiple devices in the following conditions:

- SIP device with ambiguous address
  Some devices might advertise different addresses about themselves. For example different address in Via, Contact and physical socket address. In these situations the VoIP server might send invite to different addresses to be sure that the call arrives to the user device. This can be disabled if you set the “allowforkforsignaling” global setting to 0
- User registered from multiple locations
  The VoIP servers allow users to be registered from multiple locations. For example a user can register with the same username/password from both of their laptop and smartphone. In this circumstance the server will send incoming call to all devices and once the user pick-up one of the device, all the other calls are canceled
- For simple call forward when the target is offline or doesn’t answer, you can just use the “On No Answer” setting frim Users and Devices -> Functions tab.
- User ringgroup (“Users and devices” form / tb_users.ringgroup)
  User devices can be configured to ring in different locations (by filling tb_users.ringgroup with phonenumbers/usernames separated by comma). Single line extensions are supported.
  The calls can be also serialized with the RGMode option: 0=forked call (default), 1=round robin
- Routing ringgroup (“Routing” form)
  Calls can be routed explicitly to multiple locations also from routing. Define a pattern (for example incoming calls to 12345) and add multiple endusers for the route (right side of the routing form) and select the “Is Ringgroup” checkbox (if the checkbox is checked then the call will be forked to all destinations at once; otherwise it will be routed to the highest priority destination or to the “best” one)

Call forking can be achieved in 2 ways:
-Using separate sessions (separate endpoints with their own unique SIP call-id)
-Via branch (same endpoint sending to different address using different via branch values)
The first session receiving 200 OK is the winner. For the others a CANCEL is sent.

Note: there are also other ways to create multi-session calls such as call rerouting, p2p, forwarding, transfer or conference.

4.11.3. Caller ID

You can setup the caller id display/hide (CLIP/CLIR) function by changing the **CLI** setting for the actual user.

The following values can be used:

0: forward always   (forward asserted as normal number always!);
1: normal handling (forward asserted as normal number) –default;
2: forward as asserted identity always (identityrewrite asserted);
3: forward as asserted identity only to trusted domains (identityrewrite asserted);
4: normal hide (no idenityrewrite forwarding) ;
5: force hide (no asserted identity too!).

The **SIP Privacy** method is also supported.

4.11.4. DTMF

H323 gateways support the following DTMF send/receive methods:

-as Q931
-as String
-as Tone (In-Band)
-as RFC2833

SIP endpoints support the following DTMF send/receive methods

-INFO method
-as Tone (In-Band)
-as RFC2833

DTMF in GSM network (send/receive):

-InBand DMTF

The SIP Softswitch (server) will parse DTMF received:

-INFO method
-as RFC2833
-convert from the above modes to “in-band” dtmf if needed

Usually In-Band DTMF are supported by any vendors in endpoint devices but will not be parsed on IVR servers because of high computation requirements to decode the encoded RTP channels.

The following DTMF global configuration settings are defined (key,default value, comment):

- defdtmfpayload = 101; //default payload number can be also negotiated between the peers
- maxdtmfdigit = 16; //max dtmf digit number (used in SDP)
- ivrdtmfmethod = 4; //0=inband,1=info,2=inband+info,3=RFC2833,4=RFC2833 and info,5=inband, RFC2833 and info,6=inband, RFC2833 or info
- dtmfdigittimeout = 6;
- deftmfsleep = 500; //msec delay on space w or W chars
- deftmfdelay = 0; //msec delay between 2 dtmf digit (default is 0. you might increase it usually not more than 1000 msec)

- deftmfpackettime = 0; //0 means default value in msec (after the negotiated codec) usually between 10 and 60 msec
- deftmfpacketcount = 6; //packets to send one dtmf digit
- deftmfendpacketcount = 2; //packets to end one dtmf digit

  //the time spent for one dtmf digit is: (deftmfpacketcount + deftmfendpacketcount)*deftmfpackettime + deftmfdelay msec

4.11.5. Call Hold

The call-hold feature is implemented according to SIP specification. Most UA will support it.

4.11.6. Call Forward

You can enable the call forward feature by filling the following fields in the user configuration form (Functions tab):

forwardonbusy: telnumber where we have to forward the calls when busy
forwardonnoanswer: telnumber where we have to forward the calls when we have no answer or the called user is not online
if set to * then it will lookup and forward to "best" child user
set the "**noanswertime**" value after your preferences
forwardalways: rerouting

Call forwards are also supported by the **"Moved" (code: 300-305)** SIP message, so users can set the call forward in their phones (no need to change the "forwardalways" setting on the server)

The server behavior can be controlled by the "canmove" global config option:
0=not allowed
1=forward the 3xx message 1 to the caller (default)
2=callednumber change allowed (not safe, because pricing can change in different directions)
3=domain change allowed (not safe, because pricing can change in different domains/directions)
~~4=call routing again~~
5=call forward (call forward 301/302)

More details about call forward can be found in the wiki.

4.11.7. Call Transfer

-The standard sip calltransfer protocol is supported (**REFER, replaces methods**) using the transfer button on sip phones, but because many sip devices have problems with call-transfer, the DTMF mode transfer is supported. If the transfer fails, than the **Cisco "Also"** method is also supported for transfer failover.
-Calls can be transferred by the following **dtmf digits**:

***9*number#** -> unattended transfer
***8*number#** -> transfer with consultation

-You will hear a music if the calltransfer is in progress, or a failure notice if it failed.

-After the transfer successfully begun (ringing to the new client), you have the following DTMF options:

1: talk with the new called party when it is connected (not possible if already is connected),
for example to discuss the purpose of the caller.
When you hang-up, the caller and the new called party will be connected.
You don't need to press 1 if you have started the transfer with consultation (*8*).

2: disconnect the new called party and talk with the original caller (not possible if already transferred)
for example you can redirect the caller to a new destination

3: stay in line with the old client (you have to hang-up when you finish to complete the transfer)

*Note: DTMF ** will always reset your already entered digits to **

Related configurations:

handlerefer
Specify how to handle REFER requests.
-2: block
-1: block for outbound/allow only for endusers
0: auto/default (forward as-is or handle if REFER is not supported by peer ep or if the transfer was made with Replace)
1: handle as unattended transfer (except for transfers with Replace as in that case we might handle the Replace)
2: handle as attended transfer (except for transfers with Replace as in that case we might handle the Replace)
3: forward as a new call
4: just forward the REFER as is

fs_handletransfer
Specify how to handle WebRTC call transfer requests (applicable only for WebRTC calls)
-1: auto/default (if target user is SIP and not here and no replaces)
0: handle the transfers in the WebRTC stack
1: auto (handle the transfers in the SIP stack if target user is SIP and no replaces)
2: handle the transfers in the SIP stack whenever possible
3: always force handling the transfers in the SIP stack

allowclientinitiatedtransfer
Specify if you wish to allow transfer requests from client endpoints
-1: auto/default (it will allow to local user or if no billing is required)
0: no

1: yes

playtransfertone

Enable/disable ring or music playback on transfer.
0: nothing
1: ring
2: music (default)

discontransfer

Disconnect old parties on transfer
-1: auto (default)
0: no
1: yes on C party connect
2: immediately on succ if no refersub
3: immediately on fail if no refersub
4: always immediately

holdontransfer

Specify if call should be muted on transfer.
0: no
1: yes (default)

refertorewrite

Specify if the server should attempt to fix the Refer-To SIP header if the other parties are elsewhere (old refertoforward)
-1: auto/default
0: don't rewrite
1: auto
2: always rewrite

Technical description:

Structures:

```
calltype = eTransfer
//for the transferrer client ep
transferstate; //0=unknown, 1 = mute to all, 2=talk with old client, 3 = talk with new client, 4 = talk with both clients
because we have 2 client ep (called) for the server (caller):
                GetOldTransferClient(); //will return the operator
                GetNewTransferClient(); //will return the new client
```

Workflow:

```
Unconditional dtmf received (*9*number#)
Set Istransferrer = true
RouteCall()
        If fail than play transferfail
```

4.11.8. Three-Way Calling

Three way dialogs with call holds are supported.

4.11.9. Call Waiting and queuing

Call waiting scenarios are supported. Need to be implemented in UA to work.
Group hunting and queuing are implemented only in callcenters.

4.11.10. Call Take-Over

You can pick up or transfer an incoming or existing call to any other device, by **dialing *7*origcalled** on the target device.
You must be in the same user group or billed user for this feature to work. The **SIP replace** method is also supported.

4.11.11. Conference Calls

Conference call can be made in the following ways:

- Standard SIP conferencing
  -Three-way calls
  -Conference managed from client side (client side RTP mixer)
- Server managed conference calls via DTMF:
  Two-way conversation can be extended to multiparty by inviting new members with the following DTMF sequence:
  ***1*number#**
  Only the session initiator (original caller) can invite new members to the conversation! When the session initiator left the conference, the conference will be destroyed (other members will be disconnected too).
  This method can be very useful if your SIP client don’t have built-in conference support.
  Note: Make sure that the SIP clients are sending the DTMF digits in RFC2833 or SIP invite format (not in-band RTP)
- Conference rooms:
  If you enable the PBX extra module from the server configuration wizard, then the conference room feature will be added automatically with room access numbers 5100-5199 for narrowband and 5200-5299 for wideband conference. (The room pool can be also extended to allow up to 10000 conference rooms). You can use the “getconfroom” API to auto allocate a conference room and its access number for the user.

4.11.12. Voice Mail

To enable voicemail on your server, make sure to enable the PBX Extra module in the server configuration wizard or set the **fs_pbx** global config option to 1. Also make sure that the voicemail global config option is not set to 0 (setting voicemail to 0 will disable the voice mail feature system wide).

Enable voicemail for a user by checking the “voicemail” checkbox. (Users and devices form -> Functions tab).
At least one of the voicemail handling checkbox should be also selected (on busy, on no answer, always).

Recorded voices can be also sent to the user **email** address in wave or mp3 file attachment.
Users can access their voicemail box by calling 5000 (with pin authentication usually from external network) or by calling **5001** (with auto sip digest authentication from a SIP client).

The following config options are defined for the **voicemail** global config:

-1=auto
0=disable
1=enable
2=force
3=force also for called

You can force enable voicemail for all users by setting the voicemail global config option to 2.

Using the base voicemail module, users can also replay the last voicemail by the following DTMF digits: ***3*** or listen their voicemail trough the IVR (since this feature is rarely used, there is no mizutech support for it in the new server versions). This feature is not available in the extra pbx module, but here you have a voicemail access number.

Checklist:
-fs_pbx global config set to 1
-voicemail global config is not 0
-outgoing SMTP settings are set properly in the global configuration (by default it will use mizutech email server which you can change if you wish)
-the called user has voicemail enabled and one of the followings are enabled: voicemail on busy, on no answer, always
-the called user has a valid email address set
-for the core voicemail only: voicemailcampaign is set correctly in the global configuration (must point to voicemail_rec script which also have to be set correctly)
-for the core voicemail only: for the listening script you have to properly configure the voicemal_play script (if listening trough the IVR is required)

Another related feature is the “Missing call notification”. Just check this option to have an email about all unanswered calls.

More details can be found in the voicemail wiki.

4.11.13. Voice Recording

Call recording can be toggled on/off globally or per user.

Enable/disable globally from Config menu -> Configurations -> Recording -> Calls.
Enable/disable recording per users from the Users and Devices form -> Functions tab -> Voice recording checkbox.

Recorded calls can be listened or exported (as wav or mp3) from the CDR form.

Details:

The voice recording option can be set for any user by checking the “Record” checkbox on the user configuration form in MManage.
Conversation will be saved in the directory specified by the “serverftpvoice” global config option.
The exact location will be: serverftpvoice\databasename\currentday\voice.xxx

A separate backup can be created in the directory specified by the “voicebackupdir” global config option.
Old files can be deleted by setting the “keeprecorded” option accordingly (days to keep).
Recorded files are compressed and encrypted by default.

On new server installation, make sure that the voice directory is accessible via ftp for the MManage (for listening on the “CDR Record” form).
A separate ftp account might be enable with the same credentials as the database login.
Recorded conversation can be downloaded from the web.
Users can replay the last record by the following DTMF digits: ***4***
You might also disable peer to peer rtp routing to prevent media bypassing the server. For this set the “disablep2prtprouting” parameter to 1.

You can also use the voicerecdownload API to download any recorded file(s).
The request should look like this:
https://yourdomain.com/mvapireq/?apientry=voicerecdownload&callid=XYZ&maxrecords=1&authkey=SERVERKEY&authid=ADMINUSERNAME&authpwd=ADMINPASSWORD

In case if your are using the Mizu webphone, then the HTTP or FTP upload URI can be also configured in the webphone itself using the webphone voicerecupload parameter or voicerecord API.

Call recording will use extra disk space, can make the routing and media path longer by disabling peer to peer routing and also will increase your server I/O and CPU usage.

4.11.14. Chat Recording

The users chat history can be recorded and stored in table tb_messagelog.
For this set the "logmessenger": 0=no,1=on low load,2=always
You also have to disable fast message forwarding: set the "fastmessagequeue" configuration option to 0.
The recorded messages can be seen on the "Chat Logs" form MManage.

4.11.15. IVR

You can assign an IVR menu to any user if you set the "**ivr**" field to a valid IVR id.
IVRs are directly mapped to campaigns and can be edited in the script editor. Set your CampType to 2 if you have assigned to an IVR. See Scripts for more details.

The following actions can be defined at serverside:
Wait for DTMF: wait until match or eof characters found or timeout
Play a File: play any voice message. A repeat count can be defined. You can choose to jump to next action when the play begins, or only when finished (the client will have to listen the whole message before he can move forward with dtmf)
If you uncheck the "Wait for playback to finish", than the IVR will continue with the next action immediately and the playback will be stopped only when a "stop pending playback" condition is reached or a new playback is started.
Play SQL Result: you can write any sql query here. The first field in the first column will be played. The SQL can contain keyword, defined below.
Forward to Phone Number: forward the caller to any number.
Forward to Group: forward the caller to a group.
Forward to SQL: forward the caller to the phone number or username returned by custom sql.
Forwarding will be tried "retry" count in every "forwardretrytimer" sec.
Charge: unused (use "execute sql" for charging).
Mailbox: unused (use "execute sql" and the "record" actions to implement a custom mailbox).
Execute SQL: execute any sql command (check allowed keywords below).
Record: record conversation. Can be used for voicemail recording.
Delete Recorded: delete current recorded and start playback the next recorded audio.
Next Recorded: playback next recorded audio (leaving the current).
Prev Recorded: playback prev recorded audio.

Text playback will be handled in different ways for numeric characters and for any other character. For numbers, the internal **NumberToSpeech engine** will be used (configurable in multiple languages). For any other text the default **TextToSpeech engine** will be used (available only in English in this moment).

For **skill based callcenters** you can use the "Forward to Group" or the "Forward to SQL" actions.

Voicemail can be implemented by the “**record**” actions.

Keywords at server side:

- currentnumber
- callduration
- ringduration
- opusername
- opname
- currdatetime
- currdatetimesql
- currdate
- currweekday
- curryear
- currmonth
- currday
- currhour
- currmin
- currsec
- calleruserid
- calleduserid
- proxyid
- origcallernumber
- callernumber
- callername
- origcallednumber
- callednumber
- calledname
- techprefix
- called_norm
- auth_username
- auth_password
- transportip
- fromip
- rtprecip
- transportport
- fromport
- rtprecport

- callertype
- callstate
- cc_clientid
- cc_ccid

Technical description:

```
On routing if calleduser.ivrid > 0 than set ep ivr_id and calltype to eIvr
        No other ep will be created
        ivr_id is the campaignid
On invite (after routed)  if calltype = eIvr () call IvrAction(actIvrStart);
IvrAction(EIvrActionType actiontype)
        1. the actiontype will tell what was happened: actDefault, actIvrStart, actReady, actUserInput, actTimeout, actFail
        2. load the scripts if already not loaded
        3. Check next action todo (jumpto). Especially check for dtmf input action
                a. if actiontype == actUserInput than check if input finished (eof or maxlen)
                b. if play finished and need to repeat, than play again
                c. othervise set jumpto
        4. JumpTo next action if needed (check enterconditions)
        5. Do the next action (datainputtype, if any)
               Play file, execute sql, forward, etc
```

A more detailed description about the Mizu IVR system can be found here and here.

4.11.16. Callback number

Used for handling incoming calls by callcenter operators.
See the “callback“ global config option.

4.11.17. Callback services

To configure callback services you will need to perform the following tasks:

1. Create callback access numbers: just create callback enduser from the “Users and devices” form and specify the “Act as IVR” settings on the “Functions” tab.
2. Create a callback IVR script: callbacks always need an associated IVR to perform some actions once the server calls back the user (such as playing some announcement or request the user to enter some phone number to forward the call to)

Callback services can be implemented by various methods:

- From the enduser webportal
    - By entering the number to call on the website
    - On the ensuser webportal you can find callback and p2p page (they are almost identical)
- By CID or ANI callback

    - Setup a callback number (enduser) and assign the callback IVR script to it.Assuming that our callback number is C:
        - A call C
        - C will drop the call
        - C will call back the A number and will ask to enter the B number (this is done with the IVR. You should already have a “callback” ivr script template there)
        - Once A is ready with the number input, C will call the B number
        - Once C and B is connected, C will goes out of the path and will interconnect A with B
- By inbound SMS
    - By a special SMS message
- By API
    - Using the server HTTP API, the callback functionality can be easily integrated in any application or website

To define access numbers set tb_user.iscallback to the required ivr (campaignid) and tb_user.anumberlookup to 1.
The calls will be authenticated based on “ivrauthentication” global config value and/or “iscallback” user field. When using only A number authentication be aware of frauding possibilities.

Callback will be initiated from “callbackcallernumber” A number or from the user who received the call if “callbackcallernumber” is not specified.
The callback will automatically start the IVR specified in the “iscallback” field or in “ivrid” field of the access number.

The IVR script to be used for a call, can be loaded from the followings:

- Callback user (access number) tb_users.ivrid field (you can specify it on the “Users and devices” Functions tab)
- “ivr” URI parameter if the callback is API initiated
- Or loaded from the defcallbackivr global config if the above is not specified

Ivrauthentication is deprecated. Use the anumberhandling global configuration: 0=disabled,1=only add, 2 = only accept, 3=add and accept (default)

iscallback

- ➢ 1=act based on anumberhandling (default - connect to the ivr if not authenticated as enduser, otherwise callback immediately)
- ➢ 2=connect to the ivr if not authenticated as enduser, otherwise callback immediately
- ➢ 3=connect to the ivr if authenticated as enduser, otherwise callback immediately
- ➢ 4=drop the call if not authenticated as enduser, otherwise callback immediately
- ➢ 5=drop the call if authenticated as enduser, otherwise callback immediately
- ➢ 6= callback immediately always
- ➢ 7=always ivr
- ➢ other=ivr id to use (otherwise will check the "ivrid" field)

Example 1: if you want to call only authenticated users (no expense for failed authentication on callback):

- ivrauthentication: 1
- iscallback: 1

Example 2: if you have a free to call number or you can afford the callback payment to unauthenticated users then you can use the following settings:

- ivrauthentication: 1
- iscallback: 8

Example 3: no free callback and no trust in A number authentication

- ivrauthentication: 3
- iscallback: 2

**Note: if you would like to allow calls to IVR or to callback numbers from not authenticated parties, then set the "freeivraccess" and "freeaccessuserid" configuration options accordingly.*

Technical details

1. call initated from console, db check or iscallback user
2. mainlogic process asyncCallback and initiate call
3. client endpoint changed to server ep on connect
4. ivr starts requiring pin if needed and the target number
5. ivr callforward with 2 leg billing

4.11.18. Calling Card services

For a calling card service you must setup an IVR (and a campaign) that will handle the authentication and forward the call to the requested destination.
To create an access number, add a "enduser" and set the ivrid to the campaign id you wish to use.

You can also allow authentication based on A number by setting "anumberlookup" for the traffic sender.
Example IVR:
-Play File: Welcome to xxx. Please enter your PIN number
-Wait for dtmf (acquire PIN digits)
-CallingCardAuthentication: (Will check the pincode in the database). On success go to next item, on fail play a prompt to try again
-Play File: Please enter the destination number
-Forward to phone number [dtmftrimmed] (call forwarding)
-Finish (release ivr)
If you want to bill the user for calling the ivr, then set the "ivrbilling" global config to "0". Otherwise set to "1" and only the forwarded call will be billed.
If you need 2 leg billing, then set the "ivrbilling" value to 2. In this case 2 CDR records will be generated for the original caller.

PIN codec can be stored in the user table. Their type must be set to 0, isoperator to 6 and authentication based on username (5)
The authentication can be based on username, password or username+password or depending on the "**callingcardauth**" config option.
The following values are defined:
0=pin

1=callingcard username
2= callingcard password
3= callingcard username+password
4=any username
5=any password
6=any username+password
7=any username+password or username+pin
8=username for callingcard and username+password or username+pin for other users (default)
9=pin
10=password or pin
11=all combinations
12=calling card username or enduser username + password or enduser username + pin (default)

Calling cards can be generated in batch from the Config -> Users menu.
When using the pin field based authentication, make sure that user has valid pin codes set in this field (when the users are automatically generated, the pin is set to be username+password).

4.11.19. Phone to Phone (P2P) calls

You can implement this service on your website or in softphone.
The calls can be initiated with the following methods:

1. connect to the server console (a TCP connection) and issue the p2p call command. All communication can be encrypted and the clients are authenticated properly.
2. Insert a new record to tb_conferences with type set to 3

The billed user will be the logged in user. The p2p command has 3 parameters. A number, B number and IVR id (optional).
2 CDR records are generated for these calls, both of them billed to the initiator (based on your usual pricing).
The call to the A number will be initiated from “callbackcallernumber” configurable number or from the user who initiated the call.

You must set the “checkdbconf” global config value to 1 or 2 if you need to initiate phone 2 phone calls from the database (initiated from website for example)

Note: unless callback service, p2p doesn’t need any IVR as it will just interconnect two endpoints

4.11.20. Virtual Servers

You can create up to 100 virtual servers on a single pc. These are completely separate billing/routing/signaling entities and can be used for different business models or to offer hosted services to separate companies or organizations.

If you can afford, use a specific number prefix for each virtual server and setup the routing on the main server accordingly.

If a specific number scheme cannot be used:
On the main server you can specify to wich virtual server a number belongs in the “numbers” table (MManage -> Other -> Phone Numbers). In the location field, the virtual server sipuser id must be entered. -1 means that the number belongs to the main server.

Make sure that each service will use a different IP (bind ip) or the ports are set to be different.

4.11.21. DID

Virtual/real number mapping can be used to staticallt/dynamically routed DID numbers and for A number assignments. See the “DID numbers” section under “Access” for more details. See the DID number section for more details.

4.11.23. Barge-In

Intercept call barge or call spying can be done multiple ways:
-With the Voice Here module from MManage (or just right click on a record on the Current Calls form
-Via the 5009 access number for PBX calls
-From the customized SIP clients by sending a specific SIP header (defined by the bargeinheader)

4.11.24. Unified communication

The Mizu VoIP server implements a broad range of UC features.
Most of these features are available on the server by default with no special settings.
Just make sure that you have selected the “PBX Extra” module on the “Roles and Features” section of the configuration wizard.

Here are some of the UC related features:

- Video: works by default conform SIP standards with all video capable sip endpoint
- Chat: works by default via SIP standards RFC 3428 (chat recording can be also enabled)
- File transfer: requires client side support which is rare (you can use our webphone or windows softphone for example)
- SMS: supported as sip messages (if your carrier has this support) or you need to configure an external SMS gateway for this to work. For example Clicatel. Here is a short description.
- Fax: enabled by default and both analog and T.38 are supported. (Fax to email is not supported at this moment but it can be added via a gateway)

- Conference: via SIP standard, via dtmf *1*number# or via conference rooms using extensions between 5100-5199 for narrowband and 5200-5299 for wideband (these can be changed)
- Voicemail: enable from users and devices form -> functions tab. Default access number are 5000 (with pin) and 5001 (with auto authentication). Auto-email forward is enabled by default.
- Many Others: A lot's of other UC related features are implemented. Check the configuration wizard, global configuration and the documentations for more details

4.11.25. Other features

The Mizu VoIP server implements most of the SIP related RFC's as described here.

4.12. Additional modules

The following modules can be enabled optionally and are part of the "all-in-one" package:

4.12.1. Calling-Card

The calling card functionality is implemented by using the IVR module described in the previous section

4.12.2. SMS

Users of the Mizu VoIP server can send and receive SMS messages from softphones, web portal or API.
The server is capable to also convert simple SIP text messages to SMS if the target is a mobile phone number.
Incoming SMS requests can be converted to HTTP(S) or SMPP(S) requests and forwarded to an SMS gateway defined by the smsurl global config value.
The server can also use SMS for special purposes such as SMS code based sign-up verification (uauthverify) or administrative alters.

User interface

Once SMS is configured on your server, the endusers has various method to send SMS messages:

- Using the API. This can be easily integrated with any custom user interface or third party tool. See the API section below for the details.
- Customized softphones: softphones provided by Mizutech can have a "Send SMS" feature
- Third-party softphones: the server is cabable to convert standard SIP chat messages to SMS if the target is a mobile phone number, thus SMS can be sent from any SIP endpoint (sendsmsmessages global config option enabled by default)
- Web portal: Users can also send SMS messages from the enduser web control panel (SMS page)
- Server console: you can easily test SMS messages using the asendsms command. Example: asendsms,TONUMBER,Hello word

SMS Integration

To be able to offer SMS capabilities for your users, you will need to interconnect with an SMS service provider or with an SMS gateway.
The Mizu VoIP server has support for SMS over various protocols such as SIP IM, email, 3GPP SMS routing, SMPP or HTTP GET/POST with proprietary API's using clear text, JSON, XML or other text based payload types.
You can configure integration with any SMS service as described below.

Outbound SMS

The server can connect to SMS providers (such as clickatel , routomessaging or sight) and send SMS message via SMPP(S) or HTTP(S) requests.
If you don't have an SMS provider yet then please consider this list:

clickatell.com
www.sinch.com
infobip.com
messagingbay.com
www.routomessaging.com
hqsms.com
tm4b.com
smsbts.com
bulksms.co.uk
smsxchange.com
textmagic.com
truesenses.com
txtlocal.co.uk

We recommend clickatel and sight due to their affordable pricing and system stability.

The SMS API URL and it's parameters must be obtained from the SMS provider. This is usually a HTTP link or SMPP config.
Set the URL in the global configuration under the smsurl key.
If the SMS is to be sent over SMPP, then the URL must have the following format: SMPP;key1=value1;key2=value2;…etc. See the parameters below at the "SMPP parameters" section.
If the request have to be a HTTP POST (instead of GET) then you can set the data to be posted with the smssendata parameter or stored in the smssendata.txt file in the server app folder.
If some special HTTP header(s) are required (for example Authentication) then you can specify it with the sms smssendheader parameter or in the smssendheader.txt file.

You can use the following keywords in the smsurl, smssendata and smssendheader:
[fromid], [fromnum],[tonum],[message],[smscounter],[smsid],[gatewayid],[clientip] or any SQL query in brackets such as {SELECT …}.
These will be replaced automatically to their respective values for each session.

The delivery success is guessed by default from the response or you can configure the smssuccessstring and/or smsfailstring the keyword to look for (the response is parsed as simple text).
Unicode messages are also supported.

If you wish to route to SMS messages to multiple providers then you should follow the following steps:

- set the smsrouting global config option to 1
- add SMS GW users
- setup the routing
- setup billing
- HTTP POST data will be loaded from smssendata_ID.txt loaded file (where ID is the gateway id -tb_users.id)
- HTTP headers can be added from smssendheader_ID.txt file (where ID is the gateway id -tb_users.id)

See the “SMS Routing” section below for clarifications regarding outbound SMS routing.

You can find more details about SMS setup here.

Inbound SMS

You can also receive incoming SMS messages as HTTP callback requests.
To enable SMS receiving, set the smsreceive_route parameter to 4 (or any other value as described below).
For incoming SMS you will have to configure your SMS service provider callback URL to point to your server smsreceive API which looks like this:
http://domain.com/mvapireq/?apientry=smsreceive
(Replace the domain.com with your SIP server domain name or IP address and use HTTPS instead if you enabled TLS on your server)
For example a simple SMS send might look like this:
https://sip.mydomain.com/mvapireq/?apientry=smsreceive&from=1234567&to=98765432&message=Works&now=555
(You can also use POST instead of GET)
See the smsreceive_xxx parameters below for other incoming SMS related configuration options.

SMS routing

In case if you have only one SMS gateway configured globally by the smsurl or smpp configuration, then all SMS will be routed to this gateway.
In case if you wish to use multiple SMS outbound gateways, then create SMS GW users and configure routing rules for them.
The routing rules can be configured exactly as for outbound calls, the only difference is that you must set the “Type” for the routing patters to “SMS”.
Once configured, SMS messages will be routed according to your routing rules.
See the above “Outbound SMS” section for more details.
In case if you wish to rewrite the outbound SMS number, check this FAQ point for the details.

SMS billing

After SMS messages a CDR record will be created with its type set to “SMS”. You can see them on the “CDR” form.

For unified (simplified one price) billing you can use the smsprice and the smstime global config options. In this case all sms messages will be billed with the same price, regardless of the destination). The smstime is set to -1 by default which means 60 seconds for up to 160 characters and then an addional 60 seconds for every 140 characters in excess.

Otherwise for normal SMS billing, the prices can be set on the “Price Setup” form the same way as for voice calls.
The only difference is that all SMS messages appear with be accounted wiht 60 sec duration by default and you must set the “ServiceType” to “SMS” for you SMS related billing packets.
If no pricing found, then the sms messages will be billed after the smsprice global config value.

Note: CDR generation for SMS messages might be delayed by about 10 seconds (because the server also sends a second requests to verify wether the SMS was sent successfully or not).

Rewrite SMS number prefix

Create a stored procedure named v_dialplansms with the following inputs:

- tonum varchar(256)
- fromid int
- fromnum varchar(64)
- clientip varchar(64)

This sp must return one field containing the rewritten target number (“tonum”)
Set the “usesmsprefixsp” global config option to “true” and reload.

API

The following API entries are defined:
sms: old deprecated
sendsms: asynchronously (fast) send SMS for endusers (check user rights and billing)
sendsmssync: synchronously (wait for delivery result) send SMS for endusers (check user rights and billing)
asendsms: asynchronously (fast) send SMS for admins
asendsmssync: synchronously (wait for delivery result) send SMS for admins

API parameters:
Sender number: anum or from
Target number: bnum or to
Message text body: message or txt

API Answer:
On success, the API will answer with message beginning with OK text.
On failure, the API will answer with message beginning with ERROR text.

By default the "asendsms" and "sendsms" API will send the SMS asynchronously. This means that you will receive a positive answer immediately once the send was initiated (which is then completed in a separate thread).
You can use the "asendsmssync" or "sendsmssync" API to send SMS synchronously. This will wait for the final status.
We recommend the "asendsms" and "sendsms" API to be used for high volume SMS since it is much faster. The "asendsmssync" and "sendsmssync" API should be used only if you need an accurate result (delivery succeed/failed) for the API. Otherwise you can always check the CDR for the delivery results.

Example:
http://yourserveraddress.com/mvapireq/?apientry=asendsms&authkey=xxx&authid=xxx&authmd5=xxx&authsalt=xxx&anum=FROMNUMBER&bnum=TONUMBER&message=MESSAGE&now=555

Configurations

All SMS related settings can be configured from the following places:

- Users and devices form: you might create multiple outbound SMS gateways as "SMS GW" user entries
- Routing form: you can create routing rule patter(s) for SMS messages by setting the "Type" to "SMS" for the directory definition
- Price Setup form: create billing rules for SMS messages by specifying "SMS" for the pricing packet(s) service type
- Global configuration: search for "sms" and/or "smpp" to list all SMS related settings (discussed here below)

The SMS related global config options are listed below:

- enablesms: enable the SMS module (default is true. Set to false to disable)
- smsurl: outbound SMS URL (it can be also an executable path instead of URL)
- smssendata: HTTP data for outbound SMS if POST have to be used
- smssendheader: HTTP headers for outbound SMS (for example Authentication header)
- smsthread: 0: no separate thread (more reliable answer), 1: separate thread (better performance)
- smsverifyurl: if a second request is request to verify the SMS success (currently harcoded to use "messageId" from the answer
- smssendataverify: HTTP POST data for verify (otherwise HTTP GET will be used)
- smsverifydelay: milliseconds to wait before calling the verify URL
- smssignature: to add a "signature" text after all outbound SMS
- smsadv: set to true for advertising –to insert and advertising message before each outbound SMS (tb_advertisment)
- smsadvprefix: insert a text before each message
- smsadvvalue: users can receive some credit if advertisement are sent to them
- smssuccessstring: success keyword to look for in the response (depending on your SMS provider and payload type)

- smsfailstring: failure keyword to look for in the response (depending on your SMS provider and payload type)
- smsrouting: if SMS routing have to be used (“Routing” form): -1: auto guess, 0: global config only, 1=multiple sms gw
- sendsmsmessages: specifiy wether to convert SIP IM to SMS. 0=no,1=to mobile only (default), 2=to all phone number, 3=to all destination
- smsprice: default SMS price for a 60 second SMS (which is the default “duration” for an SMS)
- smstime: SMS messages will be treated with this duration. The default -1 means 60 seconds for the first 160 characters and an 60 for any additional characters
- smsreceive_route: how to forward incoming SMS to users: 0: disable, 1:chat,2:email,3:chat or email,4:chat and email
- smsreceive_ip: allow incoming SMS only from this IP address or IP address list separated by comma (if empty, then it will allow from anywhere)
- smsreceive_check: any string to check if the message is incoming sms, otherwise will be dropped. usually empty
- smsreceive_from: extact sms sender number from the incoming message. it can be a single word parameter name or fromstring1,fromstring2,tostring1,tostring2
- smsreceive_to: extact sms target number (our user) from the incoming message. it can be a single word parameter name or fromstring1,fromstring2,tostring1,tostring2
- smsreceive_message: extact the SMS message body from the incoming message. it can be a single word parameter name or fromstring1,fromstring2,tostring1,tostring2

SMPP parameters:

SMPP parameters can be set in the global configuration with an “smpp_” prefix (for example “smpp_server”) or inside the SMS URL as key=value pairs.
Example URL or smpp_allparameters:

- Basic: SMPP;server=11.22.33.44;username=smppuser;password=xxx;serverport=2775
- Advanced: SMPP;server=55.66.77.88;username=smppuser;password=xxx;serverport=2785;version=50;timeout=30;msgpriority=3;servicetype=CMT;SourceTON=1

Possible parameters:

- smpp_allparameters: You can store all the below parameters in this single field using the above format
- server: SMPP server address to connect to. The instant messaging server must contain a valid SMPP service application address. Usually an IP address.
- username: User ID value to be used for identification with the SMPP service
- password: This is the user's password
- serverport: SMPP server port (default is 2775)
- version: the SMPP version to be used. Possible values: 33: v.3.3, 34: v.3.4, 50: v.5.0. Default is 34 (v.3.4)
- secure: set to 1 if you need SSL/TLS connection
- systemtype: Some SMS servers might require suplying the system type which is usually an abbreviation string of maximum 12 characters. Default is empty.
- servicetype: type of service. Can be set globally or by message. Possible values: 0: default, 1: CMT, 2: CPT, 3: VMN, 4: VMA, 5: WAP, 6: USSD, 7: CBS. Default is 0.
- msgpriority: message priority. Can be set globally or by message. Possible values: -1: default, 0: low, 1: normal, 2: high, 3: urgent. Default is 1 (normal).

- msgexpire: optional MC expiration time in YYMMDDhhmmsstnnp format. p set to "R" means relative. (validity period of the current message). Default is empty. t means tenths of a second (0-9). nn means quarter-hour time difference between local time and UTC time (00-48)
- timeout: timeout in seconds for inter-packet inactivity. Default is 60. You might set the AbsoluteTimeout config to true to interpret this as total timeout.
- recipients: one destination phone number or multiple destinations separated by ; (max 255)
- recipienttype: 0: normal SMS number or ipv4, 1: namelist
- isdata: set to 1 if you wish to send the paypload as data instead of a message. Only one recipient is supported in this case
- extraconfig: additional exta configuration parameters as key=value;key2=value2;
  - SenderAddress: the SMPP protocol allows an External Short Messaging Entity (ESME) to specify its address, whether it is a phone number or an IP address. If SenderAddress is not set, the component will default to the value in LocalHost
  - SourceTON: Type of Number for the ESME. 0: unknown, 1: international, 2: national, 3: network specific, 4: subscriber number without prefixes, 5: alphanumeric, 6: abbreviated
  - SourceNPI: Number Planning Indicator for the ESME used to specify the numbering plan. 0: unknown, 1: ISDN, 3: data, 4: telex, 6: landline, 8: national, 9: private, 10: ERMES, 14: internet, 18: WAP. Default is 1 which since mobiles are usually covered by ISDN
  - DestinationTON: Type of Number for the destination ESME. Possible values: same as for SourceTON.
  - DestinationNPI: Number Planning Indicator for the desination ESME. Possible values: same as for SourceNPI.
  - DataCoding: data encoding mechanism to be used for the current message. Possible values: 0: MC Specific encoding, 1: IA5 (CCITT T.50)/ASCII (ANSI X3.4), 2: Octet unspecified (8-bit binary), 3: Latin 1 (ISO-8859-1), 4: Octet unspecified (8-bit binary), 5: JIS (X 0208-1990), 6: Cyrillic (ISO-8859-5), 7: Latin/Hebrew (ISO-8859-8), 8: UCS2 (ISO/IEC-10646), 9: Pictogram Encoding, 10: ISO-2022-JP (Music Codes), 11: Reserved, 12: Reserved 2, 13: Extended Kanji JIS (X 0212-1990), 14: KS C 5601,
  - HexString: a hex-encoded binary string to be sent to the current recipient
  - InBufferSize: size in bytes of the incoming queue of the socket.
  - OutBufferSize: size in bytes of the outgoing queue of the socket.
  - AbsoluteTimeout: determines whether timeout means inactivity timeout or absolute timeout
- ssl config (to be set only if you need to use secure SMPPS):
  - sslaccept: set to "ANY" or "ALL" to accept any certificate presented by the server. set to - to cler it and force cert validation.
  - sslencodig: the PEM/base64 encoded SSL certificate
  - sslcertstore: name of the certificate store for the client certificate. Possible values: MY: personal cert store, CA: authority certs, ROOT: root certs, SPC: publisher cert, filename: PFX or java cert store or cert+private key file
  - sslpwd: password for the certificate store (if any)
  - sslstoretype: type of certificate store for the client certificate. Possible values: 0: user, 1: machine, 2: PFX file, 3: PFX blob, 4: PEM cert and key
  - sslsubject: subject of the certificate used for client authentication. can be set to * for any

Debug

In case if you encounter any issue with SMS sending, you can find out the problem following these steps:

1. Verify the SMS related configuration described above. Search for “sms” in the global config on the “Configurations” form.
2. Note that the Mizu server SMPP module will bind as a transmitter. Receiver and trasreceiver bind is not supported.

3. SMPP connectivity and message delivery can be tested with the mnetutils.exe (found in your VoIP server app folder).
   You can also test with third party tools such as SMPPCli.
4. Check the SMS CDR records (CDR form). Delivered SMS messages has a positive “duration” set after the smstime global config (-1 or 60 by default).
   Failed SMS messages has 0 duration and you can find details in the disconnect reason and comment fields (with “All Fields” checkbox checked).
   Note: if a separate API request is used for SMS verification then the CDR records are generated with around 6 seconds delay.
5. If the problem is not clear from the CDR, have a look into the server logs (MizuManage -> Control -> Logs -> Show -> Log Folder). Make sure that the logs are enabled (Set to “On” or loglevel is set to 5) and search for “sms” in the logs (or for “sendsms” or for your SMS number or message text).

SMS CDR

The fields in the SMS call detail records means the followings:

- realduration: 0 if not sent, 60 or GetSMSTime if sent
- discreason: GSM, Normal, unspecified if sent, other on fail (the discreason code might be different depending on the ack/fail stage)
- connecttime: 1 (+send time which should be 0)
- discparty: 3;
- callerid: sender user id
- callername: sender number
- callernumber: sender number
- calledtype: 12
- gatewayid: SMS gateway id
- simid: smpp ourid
- calledline: 99
- simpos: 99
- callerip: sender IP addrsss
- calledip: gateway address or URL or settings
- calledname: URL
- origcallednumber: original target number
- callednumber: target number
- calleddialed: target number
- rtpsent: 10
- rtprec: 0 on error, otherwise 1
- rtpframes: smscounter (ignore for smpp)
- rtpcodec: SMS
- web: smpp protocol id (as received from the SMPP server) or our id or retval
- comment: details about the send process

4.12.3. SMS callback

Using SMS callback you can initiate VoIP calls with SMS messages.

If you subscribe to an SMS provider (like clickatel) the server can receive SMS messages (usually on http port 8084) and initiate actions regarding the content.

By sending an sms messages users can initiate callback, register a new A number for CLI authentication, initiate phone to phone call or add credit to your account.
There are different formats for all functions (short/long - depending on the authentication method). The server can use A number based authentication for the incoming sms messages. If the CLI is hided, or the A number is not specified then the users have to send their username and password or a pincode.
Instead of SMS callback we can recommend the phone to phone functionality. (In this mode the users don't have to work with the IVR. the 2 phone number can be interconnected immediate)

The authentication of the requests is done based on "callingcardauth" global setting.
Pincode can represent the concatenated username and password (if allowed by callingcardauth)

The following formats are defined:

general
username password number1, number2
pincode number
number
pincode
b (pincode based authentication. the A number will be called back)

general requests will result in a p2p call when at least 2 number is known. If only one number is know, than it will result in generating a callback to that number. Two numbers can be supplied, one by sms and the another number can be known from the sender CLI.

callback:
b username password number (username and password based authentication and the number to be called)
b pincode number (pincode based authentication and the number to be called)
b number (pincode based authentication only the number to be called have to be sent)
b (pincode based authentication. the A number will be called back)

register new number:
n username password number
n pincode number
n number

phone to phone:
p username password number1 number2
p pincode number1 number2
p number1 number2
p number2

add credit:
c username password rechargecode number2
c pincode rechargecode number2
c rechargecode number2
c rechargecode

Example http request:

```
POST / HTTP/1.1
User-Agent: Clickatell MO Callback
Host: 99.99.99.99:8084
Pragma: no-cache
Accept:
Referer: 99.99.99.99
Content-Length: 157
Content-Type: application/x-www-form-urlencoded

api_id=3144985&from=3611111111&to=3622222222&timestamp=2009-07-04 21:39:40&text=Test+1+2+3&charset=ISO-8859-1&udh=&moMsgId=5jv45t5tb42bavu34drftr
```

For callbacks you must define the campaign which will have the proper IVR content by the **defcallbackivr** global configuration setting.

You can set a httpsmscallbackip parameter if you wish to enable callbacks only from a specific IP address.

4.12.4. Web portal

The Mizu VoIP server web interface can be used by the VoIP users to perform the common tasks related to their accounts.

The web interface is handled by a separate service called “MizuWebService” with its own embedded webserver (http/https server).
Different type of user can login and will have their own different interface after login: admin, reseller, enduser and callshop interface.

You should be able to access your web interface by typing the following URI in your browser:
From a local browser:
http://127.0.0.1 or http://127.0.0.1:8080

From a remote server
http://domainname.com or http://ipaddress.com or http://domainname.com:8080 or http://ipaddress.com:8080

By default the webportal will listen on port 8080 which can be changed at any time by the configuration settings.
Make sure that the IIS (or any other webserver) is turned off if you are using the HTTP default port (80).

Recent versions are using the universal port (usually set to 80) to allow access all services via a single port, so you can access the webportal also via this port like: http://serveraddress/webvoip

If you enabled the TLS functionality, then use HTTPS instead of HTTP: https://serveraddress/webvoip

Separate functionality for endusers, resellers, callshops and admins
Functionality:
-new enduser registrations
-status
-user acc details
-tariffs
-CDR’s
-statistics
-initiate payments
-endusers: recharge
-endusers: setup call forwarding
-endusers: initiate phone to phone calls and callbacks
-enduser: send SMS
-endusers: launch webphone
-resellers: create sub reseller and enduser accounts
-resellers: create own tariffs
-resellers: generate pins and users
-callshop: watch cabin’s status
-admin: manipulate users
-admin: global statistics

To enable the SMS,phone to phone and the callback functionality the “checkdbconf” must be set to 1 or 2. (can be done via the configuration wizard by enabling the database interface)

4.12.5. Resellers

To activate the reseller module, you must set the “resellerbilling” global config option to true. This can be done also by using the configuration wizard from the Config menu.

Resellers will be able to log-in on the web portal to manage their accounts: set pricing, create endusers (and sub-reseller) accounts, see their traffic (including their CDR records) and other functions.

User parent-child relationships:
When the reseller option is enabled, you should always create one (or more) owner account.

1. below the owner account(s) you can put the “top resellers”.
2. below the top resellers you can have any sub-resellers (unlimited level - limited by your hardware resources and the “maxresellers” global config options)
3. below the reseller you can put the Enduser accounts.
4. below the Enduser account you can have sub-user accounts (attached phones, cabins for callshops, etc)

This parent-child relationship can be analyzed by using the “Ownerships” form.

Before to add any reseller in the system, make sure you have a **public reseller price** listing (end-user costs). Read the billing documentation for more details regarding reseller account billing.

Top reseller accounts can be created from the MManage application using the “Users and devices” form. You can **create “Reseller” type users** (type is 1). Then this top resellers can login on the **web interface** and create their own billing and add sub resellers and/or child endusers. Reseller can create a “base tariff” and other tariffs assigned to individual users. Multiple packets are allowed and packets can be assigned to other users or resellers directly from web (in the same way like on the “Users and Devices” form -> “Billing” tab -> Billing packet setting. Resellers usually will create their own price lists by cloning an existing list or their base tariff list. Individual reseller prices are stored in tb_billsources with their “resellerid”. If reseller has not tariffs, than the billing will be done after usual enduserprices. If reseller has no credit, the billing will be done after the provider prices (usually generating profit loss for the reseller)
set parent for top reseller accounts to an owner account

CDR:
When the reseller option is enabled, the main cdr record will store the top reseller user id in tb_cdrs.resellerid and the “othercost” field will contain the payment from the reseller (loaded from public reseller price).
Individual reseller cdr records are stored in a separate table called tb_cdrresellers (when a reseller will login to the web interface, all reports are loaded from this table and not from the main cdr table –tb_cdrs)

In the first cdr:
-the caller enduser name

-reseller field set to top reseller
-pricing after parent reseller

You can see the reseller account **statistics** by using the “Statistics” form by checking the “OC” and the “PR”/”PFR” fields or if you login with their own credentials on the web (or with parent user credentials)

Billing
There are two options for reseller billing controlled by the “**defearlypay**” global config option:
The possible values for defearlypay are the followings:

- 0=configurable (default). Early billing if the “stage” field in tb_users is set to 9. Otherwise late billing (default is 8 for the “stage” field which means late billing)
- 1=always late billing
- 2=always early billing

1. Late billing

The credit for the reseller and enduser will be deducted after each call.
(Resellers can assign any amount of credit for the users, but the calls from the users will be rejected when the credit for the reseller reach to 0)

2. Early billing

Credit from the reseller account is deducted when he add credit for the users (create new users with some default credit, increase user credit, generate recharge PIN’s).
Tariff edit is not enabled by default in this mode unless you set the allow_tariffedit or the user rights field (in thus case the resellers could create low price tariff lists so their users could call in excess)

<u>Other configuration options related to billing:</u>

- allow_createresellers = -1; //-1: default, 0=no,1=yes (allow resellers to create sub resellers)
- allow_tariffedit = -1; //-1: default, 0=no,1=yes (allow resellers to edit tariffs. This ca be influenced also by the tb_user.rights field)

If defearlypay is set to 2 and allow_createresellers is left with the default value (-1) then tariff edit is forbidden by default.

You can also edit reseller rights by user via the "rights" field in the users (tb_users) table if the allow_xxx fields above are left with the default values (-1) (On the "Users and devices" form, click on the ... button and type "Rights". Then modify it after your needs for the resellers):

- 8: can’t edit tariffs and can’t add resellers
- 10: can’t edit tariffs

- 11: can’t create tariffs with lower price
- 12: can’t add resellers
- 30: can add resellers and tariffs

Other options:

- Creditelapseunit
- Maxcreditelapsedays
- Defaultcredit
- feature_creditcards
- feature_transfercredit
- feature_creditedit
- feature_creditfornewusers
- currencyprecision
- feature_seeowntariffs
- feature_createtariffs
- feature_recharge
- feature_hasbasetariffs

Reseller programs can work in two different ways:

1. The reseller can generate PIN codes regardless of the balance in the account.
When a call is made, both the end user and the reseller are charged.
When the reseller's credit ends, the end user will no longer be allowed to make calls.

2 The reseller can generate PIN codes only if he has enough credit in the account.
If you opt for this method, you have to enable the early payments for the reseller.
You can find this option on the reseller's billing tab.

Please be aware that once you enable the early payments the reseller will no longer be charged for calls.
You might lose money if the reseller already generated PIN codes using the first method.

4.12.6. Call-shops

-callshop owners are endusers
-cabins are sub-endusers where the parented and the billed user points to the callshop owner

-callshop owners (endusers) can add/modify their sub-endusers (cabins) from the web
See the wiki articles for more details.

4.12.7. Softphone

With the “all in one” platform you can have your own customized softphone.
For a basic softphone customization the mizutech support will need at least the following details:
-brand name (name of the executable, install package and shortcuts)
-company name (displayed in various location for example in the about box)
-company logo (if any)
-icon
-voip server ip or domain
-a long list of other configurations are also supported

You can also configure and build your softphone from here.

4.12.8. Webphone

The webphone is a SIP library and a customizable SIP client running in browsers.
All the details can be found from here.
A demo version can be downloaded from here.
You can start to work with this version at any time, test and customize it then with the “all in one” platform the mizu support will send your own build that can be used in production.

4.12.9. GSM/SIM Platform

Deprecated!
Skip this chapter if you don’t have VoIP-GSM gateways

SIM Packets
Id: database primary key. Autoincrement
Provider, type, subtype: the name of the packet
Owner: simowner in case of simpackets
Allowedpartners: applied when it is a simpacket
AbsPriorityPartner: this partner will have big priority on sims that belong to this packet
PriorityPartner: this partner will have increased priority on sims that belong to this packet
NopriorityPartner: this partner will have lowered priority on sims that belong to this packet

Filering: determines how we check the blacklist and the known numbers

0-no filter: allow all numbers

1-allow blacklist „sure" level: 0,1 and 2 (tb_blacklist)

2- allow blacklist „sure" level: 0 and 1

3-allow only blacklist „sure" level: 0

4-block all blacklist

5-allow only knownnumbers (listed in tb_knowngoodnumbers)

6- allow only knownnumbers that are 100% ok (sure is 1 in tb_knowngoodnumbers)

Dialplan:

0: international number format with 00... (e.g.: 003630xxxxxxx)

1: international number format with +... (e.g.: +3630xxxxxxx)

2: area code + number (0630xxxxxx, 061xxxxxxx)

3: shortest possible number (xxxxxxx in the same simpacket or 0630xxxxxxx in other simpacket)

4: correct it to the most appropriate format if original is not correct

WaitAfterCall: how much time must be elapsed between calls to simcard belonging to this packet

MaxMonthlyMinutes: we don't route more than MaxMonthlyMinutes to simcards belonging to this packet

MaxMonthlyMinutesPeak: maximum allowed minutes in peak time / month

MaxMonthlyMinutesOffPeak: maximum allowed minutes in offpeak time / month

MaxMonthlyMinutesWeekend: maximum allowed minutes in weekends / month

MinMonthlyMinutes: this packet will run on higher priority until the min minutes is reached

Price: default minute price if not set in tb_packetprices (deprecated. Loaded from pricelist from v.3.2)

BillingStep: second increments (used for calculation simcards minutes –daily, weekly, peak, offpeak)

MinAmmount: min billing seconds (used for calculation simcards minutes –daily, weekly, peak, offpeak)

FreeAmmount: free speech seconds (used for calculation simcards minutes –daily, weekly, peak, offpeak)

MinCreditOnRoute: if the sim has less credit, then we don't route call to it

MinCreditOnCharge: if the sim has less credit, then we begin trying to charge it

Prepaid: 0=postpaid, 1=prepaid

SendFakeSMS: we send dummy sms on this sim

CanCallEachOther: the simcard in this packet will call each other periodically to generate incoming traffic

IncludeVAT: used when credit message information are received from simcards (typically via SMS) and the simcards credits are calculated without the VAT value

Currency: used when the credit messages received needs to be converted in native currency ("currency" global setting) format. If the currency is not the same as the native currency and the "convertsimcreditcurrency" global setting is set to true, than the received credit value is converted to the native currency, based on "Currency Converter" settings, found in MManage under the "Billing" section

MaxAlloc: helper settings when automatically alocating channels for a direction. (Depending on reserverfor simcard setting).

Here you can define the maximum count of simcards that can be reserved for the actual packet. Set to 0 to disable rezerving from that packet.

Credit Request Command: the command used by the server for sim credit request (used for recharge automation)

Credit Charge Command: the command used by the server for sim credit charge (used for recharge automation)

The request and the charge command must have the following syntax: <DTMF,action,simid,"message",telnum>

The "chargecode" string in the message will be replaced with a valid code if found.

You can introduce delays by inserting '#' characters in the message.

The action parameter can be

-0: used to send USSD messages

The message parameter must have the following format "AT+CUSD=command" where command is the ussd string.

Example: DTMF,0,simid,"AT+CUSD=1,*121*chargecode#"

-1: will send the specified message to the engine. The message can be any valid AT command

-2: will dial the specified telnum, and then send the message as DTMF.

If the message string if empty, than only will dial the requested telnum, hold a little and then drop.

Example: DTMF,2,simid,"",172

-3: reserved for future usage

-4: will send the specified message as SMS to telnum

Gateways

Used to configure your Mizu VoIP-GSM gateways.

The fields are the same as listed in section 4.3.1

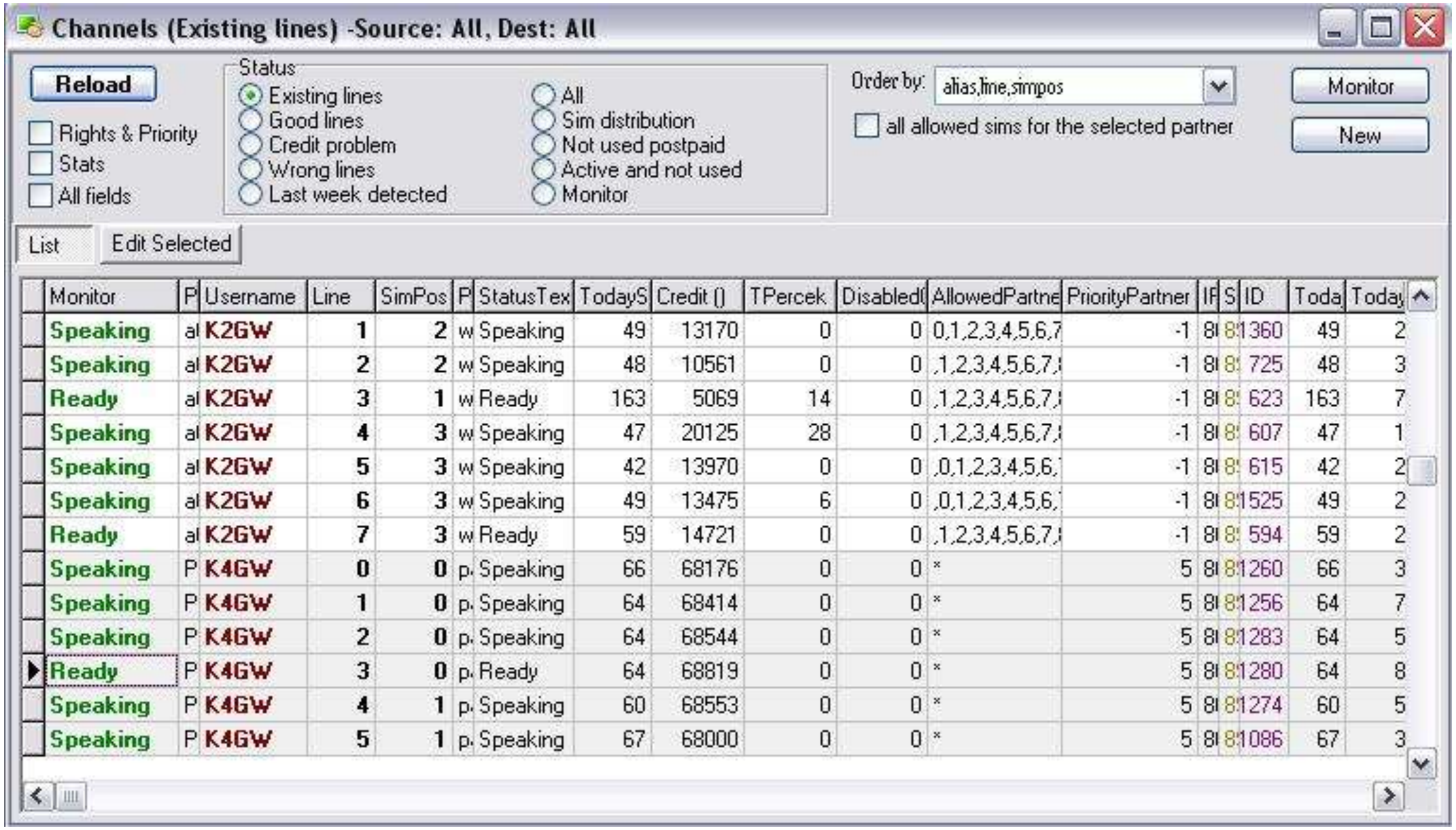
Engines

Listing of gsm channels. The fields are self explanatory.

GSM Channels

Skip this if you are not using VoIP-GSM gateways.

Usually this is the most frequently used form by the technical support. You can see the status of each gsm channel on your gateway(s).



Channels (Existing lines) -Source: All, Dest: All

Monitor	P	Username	Line	SimPos	P	StatusTex	TodayS	Credit ()	TPercek	Disabled(	AllowedPartne	PriorityPartner	IF	S	ID	Toda	Today
Speaking	a	K2GW	1	2	w	Speaking	49	13170	0	0	0,1,2,3,4,5,6,7	-1	8	8:	1360	49	2
Speaking	a	K2GW	2	2	w	Speaking	48	10561	0	0	,1,2,3,4,5,6,7,	-1	8	8!	725	48	3
Ready	a	K2GW	3	1	w	Ready	163	5069	14	0	,1,2,3,4,5,6,7,	-1	8	8!	623	163	7
Speaking	a	K2GW	4	3	w	Speaking	47	20125	28	0	,1,2,3,4,5,6,7,	-1	8	8!	607	47	1
Speaking	a	K2GW	5	3	w	Speaking	42	13970	0	0	,0,1,2,3,4,5,6,	-1	8	8!	615	42	2
Speaking	a	K2GW	6	3	w	Speaking	49	13475	6	0	,0,1,2,3,4,5,6,	-1	8	8:	1525	49	2
Ready	a	K2GW	7	3	w	Ready	59	14721	0	0	,1,2,3,4,5,6,7,	-1	8	8!	594	59	2
Speaking	P	K4GW	0	0	p.	Speaking	66	68176	0	0	*	5	8	8:	1260	66	3
Speaking	P	K4GW	1	0	p.	Speaking	64	68414	0	0	*	5	8	8:	1256	64	7
Speaking	P	K4GW	2	0	p.	Speaking	64	68544	0	0	*	5	8	8:	1283	64	5
Ready	P	K4GW	3	0	p.	Ready	64	68819	0	0	*	5	8	8:	1280	64	8
Speaking	P	K4GW	4	1	p.	Speaking	60	68553	0	0	*	5	8	8:	1274	60	5
Speaking	P	K4GW	5	1	p.	Speaking	67	68000	0	0	*	5	8	8:	1086	67	3

<u>Status Filter:</u>

Existing lines: List only current running channels. (this doesn't mean that the channel is workable. We list all channels who have reported there status in the last 5 minutes)

Good lines: only workable lines are listed. (ok status and with enough credit)

Credit problem: will list the channels with low credit and when the credit request/recharge functionality doesn't work properly

Wrong lines: list all "bad" channels

Last week detected: all active simcards in the last week

All: all channels including disabled ones

Sim distribution: all existing simcards

Not used postpaid: Some simcards may not receive calls for many days due to some misconfigurations. You may check this list occasionally to be sure that all of your postpaid simcards are working.

Active and not used: Working simcard without calls on it
Monitor: simcards grouped on gsm channels. You may detect missing “holes” very easily by scrolling down this list. This listing is almost the same as in the “Line Monitor” form.

Field Explanations:
ID: database unique identifier
SIM ID: sim identification number (you can find this number written on the simcard)
IMEI: unique gsm engine identifier
Monitor: the status of the channel. The following values are defined:

-unknown: you may have to reload
-missing: no simcard detected
-sim disabled: the “enabled” property of the simcard is set to false. No calls are routed to that simcard.
-gw disabled: the “enabled” property of the gateway is set to false. No calls are routed to that gateway.
-gw missing: no status from this gateway for more than 8 minutes
-sim missing: no status from this simcard for more than 8 minutes
-sim temp. disabled: the simcard “temporarily disabled” property is set to true. You must reenable the simcard to receive calls.
-gw temp. disabled: the gateway “temporarily disabled” property is set to true. You must reenable the gateway to receive calls.
-packet disabled: : the “enabled” property of the simpacket is set to false. No calls are routed to the members of that packet.
-closed: the channel is in the “closed” status. Can be for simchange or maybe is in restart.
-failovered: call quality has dropped below the predefined values, so the sim priority is lowered
-cannot get credit: credit automation malfunction. There are simcards from which the operator may restrict the credit request if they have no credit. Also you may need to check the packet settings related to the credit request. Check the logs too.
-wrong statistics: wrong ASR or ACD in that channel in the current day
-wrong ASR: the ASR is low in that channel in the current day
-wrong ACL: the ACD is low in that channel in the current day
-expired: the simcard has reached the predefined limits (you can configure this limits in the SIM Packets form)
-low credit: not enough credit on this simcard. Check if you have enough chargecards and the credit automation is working correctly.
-autodisabled: same as failovered
-ready (in black): no calls have been routed in the last 10 minutes on that channel (but the simcard is working without problems)

Status: channel status as reported by the gateway. Can have the following values: Gateway Disabled, Off (no info), Not Active, Gateway Disconnected, Closed, Not Ready, Ready, Dialing, Ringing, Speaking, Call Ending, DTMF, Simulating Outgoing, Simulated Incoming, Routing to SIMID, Routing to Alias, Routing
Line: the number of the gsm channel (usually from 0 to 7)
SimPos: the position of the active simslot in the current engine (usually from 0 to 7)
SIM Owner: the owner of the SIM Card
PartnerID: The database ID of the owner user
~~**CanWatchPartnerID**: database id of the partner who can see this simcard in there VPC~~

Packet: the type of the SIM Card
TodaySpeachLength: the number of active minutes on the current simcard since 00:00
ThisMonthSpeachLength: the number of active minutes on the current simcard since the first day of the current month
ThisMonthSpeachLengthPeak: the number of minutes since the first day of the current month in peaktime
ThisMonthSpeachLengthOffPeak: the number of minutes since the first day of the current month in offpeak times
ThisMonthSpeachLengthWeekend: the number of minutes since the first day of the current month in weekends
Username: Gateway Alias
Credit: current credit on the simcards. Refreshed after all calls, and corrected after credit requests (VAT included!)
InitialCredit: you may save the initial credit of the simcard here
Tpercek: special field for TMobile Tminutes
AllowedPartners: comma separated list of allowed partners and traffic senders. '*' will allow all. You may restrict the access on gateway or simpacket level instead of setting it for all simcards separately. Try to use the packet "allowedpartners" setting and leave it as '*' for the simcards!
Prepaid: loaded from the packet settings (1 if prepaid, 0 if postpaid)
Datum: the date when the simcard was inserted in the database (first use)
Comment: you may place any comment here
LastError: last error message received from the gateway related to the actual simcard
LastLog: last log message received from the gateway related to the actual simcard
LastFailedCalls: the number of subsequent failed calls (not connected calls)
LastWrongCalls: the number of subsequent wrong calls (below the predefined speech length)
LastGoodCalls: : the number of subsequent good calls (above the predefined speech length)
FieldStrength: combination of last reported field strength value in percent (0-100%) and the rx quality (from 0 to 7. 9 is invalid).
Value = field strength*10+rxqual (divide with 10 to get the fieldstrength. The remaining is the rxqual)
Pin: the security code of the simcard
LastRecTime: : the date-time of the last message received from the simcards. Every channel will send status messages in every 2 minutes and on status changes
LastCallerid: the destination id of the last call attempt
LastDialedNum: the called party number of the last call on the simcard
LastCallBegin: the date-time of the last call attempt on the simcard
LastCallEnd: the date-time of the last call attempt on the simcard
Enabled: set to 0 to disable the simcards instead of deleting it
TemporarilyDisabled: you can disable the simcard temporarily for maintenance tasks by setting this value to 1
DisabledUntil: used for automatic failovering. If the value is above the current time, the simcard is in failovered state
DisabledCause: last disable cause explained
ReenabledCount: how many times have the simcard reenabled after a failover
LastReenabled: the date-time of the last reenabling operation
TodayCallCount: call attempts from 00:00
ThisMonthCallCount: call attempts from the first day of the current month

AllCallCount: all call attempts on the simcard until now
AllWrongCalls: all wrong calls on the simcard until now (speech length below the predefined value)
AbsolutePriority: if you set it higher then on other sims, all calls will be routed here primary
Priority: routing priority boost
Filtering: determines how we check the blacklist and the known numbers

0-no filter: allow all numbers
1-allow blacklist „sure" level: 0,1 and 2 (tb_blacklist)
2- allow blacklist „sure" level: 0 and 1
3-allow only blacklist „sure" level: 0
4-block all blacklist
5-allow only known numbers (listed in tb_knowngoodnumbers)
6- allow only known numbers that are 100% ok (sure is 1 in tb_knowngoodnumbers)

Co_....: fields used by server for fake call and sms simulations
BestDirection: used for automatic simallocation
BestPrice: used for automatic simallocation
EngineID: the corresponding engine (tb_engines.id)
Credit automation related fields:

CheckCredit: credit calculation or request/charge operations needed
CrequestEnabled: automatic credit request enabled/disabled (1/0)
LastCreditRequestTry: the date-time of the last credit request command issued by the server
AllCreditRequestCount: the number of credit requests
LastCreditAnswer: the date-time of the last answer to the credit request command
CreditRequestFails: subsequent failed credit request. Check the credit automation logs if this goes above 3
LastCreditRequestFail: : the date-time of the last failed credit request
ManualCreditRequestNeed: when set to 1, the server will request the credit from the simcard in 5 minutes
ChargeEnabled: automatic recharge is enabled/disabled (1/0)
MustCharge: when set to 1, the server will charge the simcard in 5 minutes
LastCreditChargeTry: the date-time of the last credit charge command issued by the server
LastChargeCardID: the database identifier of the last charge card used for this simcard
LastChargecardPrice: the value of the last charge card used for this simcard
CreditWhenCharged: the credit value after the last recharge operation
AllChargeTryCount: number of charge operations until now
AllChargePrice: the sum of the total charge card value
FailedCharges: subsequent failed charge requests. Check the credit automation logs if this goes above 3
LastChargeSuccess: the date-time of the last successfully completed charge operation
LastChargeFail: the date-time of the last failed charge operation
CreditDiffErrors: too big difference detected on sim credit reports

SIM Cards

Same as “GSM Channels”. See section 4.2.2
The first field will show the status of the simcard (Monitor). The most frequently used values are the followings:
Unknown: the last list refresh is too old. Status cannot be determined. Click on the reload button to refresh
Missing: simid not found. Corrupt entry
Sim Disabled: simcard “Enabled” is set to false
GW Disabled: gateway “Enabled” field is set to false
GW Missing: last message received from gateway is more than 8 minute old
SIM Missing: last message received from simcard is more than 8 minute old
SIM Temp. Disabled: simcard “Temporary disabled” field is set to true
GW Temp. Disabled: gateway “Temporary disabled” field is set to true
No Packet Set: no packet settings are present for this sim. You always need to set the correct packet settings for all simcards
Packet Disabled: simpacket “Enabled” field is set to false
Closed: simcard channel status is set to closed. A simchannel can be closed for different reason. Cannot register to gsm network, Sim Change, Just restarted, etc. If this status persist, check the logs for that simcard
Failovered: server has detected wrong quality on the simcard. Traffic will be forwarded to other simcards if possible
AutoDisabled: same as “Failovered”
Cannot Get Credit: automatic credit request failed. Check the credit automation log for errors
Wrong Statistics: wrong statistics for the current day
Wrong ASR: wrong ASR detected on the channel. Treshold values can be set up from the MManage -> Menu -> Settings
Wrong ACD: too small average speechlenth detected on that simcard
Expired: maximum monthly or daily speechlength limit reached (SimPacket option)
Low Credit: prepaid simcard expired
Gateway Disc.: gateway is offline or just restarting.
Not Ready: simcard is not ready for some reason. Maybe just starting. Checj the logs if this status persist
Ready: simcard is ready to accept incoming call
Dialing: outgoing call setup in progress
Ringing: ringing signal received from gsm network
Speaking: gsm engine is ringing or call in progress
Call ending: dropping the current call
DTMF: dtmf or credit request/recharge message in progress
Simulating incoming/outgoing: calls between simcards generated by the server
Routing: the call have been routed from the server, but still not arrived to the gsm gateway. If this persist, check the log for errors. Usually means firewall/NAT problems

Note: dialing, ringing and call ending messages may not be shown in the monitor depending from the gsm gateway configuration.

If the “sendallstatus” setting is set to false, than instead of “dialing” and “ringing” only the “speaking” message will be shown.

Credits

For Identification of sms and dtmf messages received from simcards that are useful for credit request and charge
Type: 0=other, 1=succ charge without credit info,2=credit start/end, 3=failed charge, 4=need charge
Msgbgn: begins with
Msgeng: ends with -used if type is 0 (replace) or 2 (end of credit), 4 (new credit. usually 0)
Priority: check order (longer messages usually first, to not include shorter) –higher values first

SIM Distribution

All simslots are listed here.
Probability values:

not sure: the simcards was seen more than one month
probably: the simcards in the last month
sure: the simcards in the last week

SIM Utilization

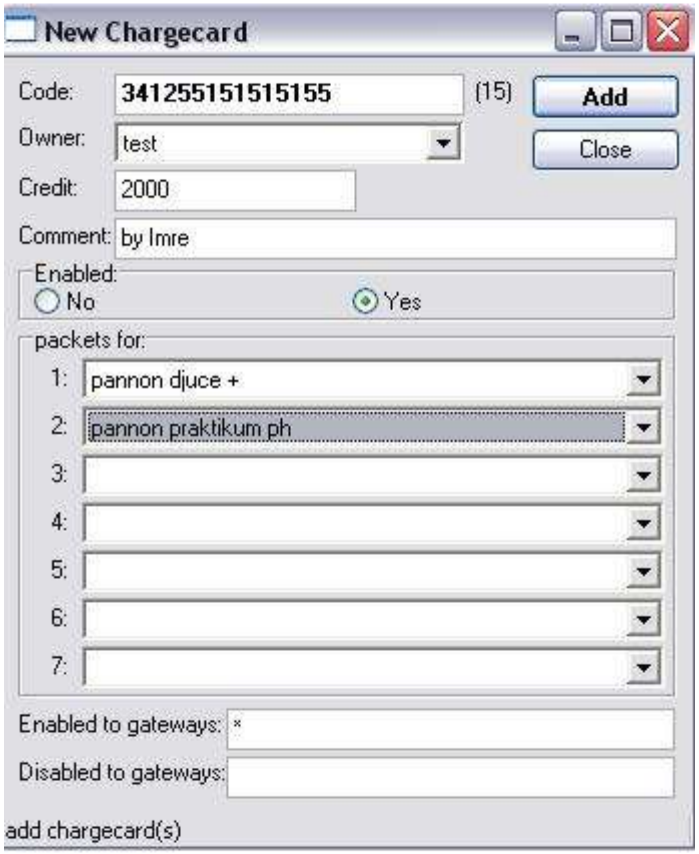
List of simcards in call duration order.

New Simcard

You can add new simcards by using this form.
However, the simcards are usually added automatically. If they are active in the gateway they will register automatically. Usually only the owner and the packet must be set manually.

New Charge Card

Add new chargecards with this form.
The charge card will be charged only on the simpackets selected (“*packets for*”) and if the *owner* will match.



Code:	341255151515155 (15)
Owner:	test
Credit:	2000
Comment:	by Imre

Enabled: ○ No ● Yes

packets for:

1:	pannon djuce +
2:	pannon praktikum ph
3:	
4:	
5:	
6:	
7:	

Enabled to gateways:	*
Disabled to gateways:	

add chargecard(s)

SIM-Bank

Skip this chapter if you are not using VoIP-GSM gateways.

Introduction:

In the SIM Bank form you can monitor the sim flying activity.
Mizu servers can automatically switch to the best simcards based on routing settings and traffic estimations.
Simcards can be bound to engines located in remote gateways, wich allows us to simulate real sim movements.
Allocation simcards between gateways is called “sim flying”.

Configuration:

The following settings are defined:

Global config:
simcallocival: minutes between sim allocations. If set to 60, than the simallocation will start at "simallocstart" minute in every hour
maxflysims: maximum simultaneous flying simcards

Gateway config:
canalloc: if automatic sim allocation and flying is allowed in the gateway (0=no, 1=only alloc,2 = fly and alloc)

SIM Packet configurations:
canalloc: if automatic sim allocation and flying is allowed with the simcards in the packet (0=no, 1=only alloc,2 = fly and alloc)
minflyday: minimum flying minutes/day
maxflyday: maximum flying minutes/day (0=disable flying)
minflyweek: minimum flying minutes/week
maxflyweek: maximum flying minutes/ week (0=disable flying)
minflymonth: minimum flying minutes/month
maxflymonth: maximum flying minutes/ month (0=disable flying)
minflyonce: minimum time to fly at once
maxflyonce: maximum time to fly at once
minstayonce: minimum time to stay (not fly) at once
maxstayonce: maximum time to stay (not fly) at once
minflydeep: minimum gateway count to fly trough at once
maxflydeep: maximum gateway count to fly trough at once

Simcard fields:
origgatewayid: original gateway for the simcard (gatewayid can be the virtual gateway)
origline: original line for the simcard (line can point to the virtual engine)
origsimpos: original position for the simcard (simpos can be the virtual position)

lastpresent: last seen (date-time)
presenterrors: simcard should be active but it is not active (counter)
lastpreserror: last not found time

flyetosim: the current engine must bind to this sim (simid): -2 = err, -1 = not set, other = simid for the engine

todayfly: today flying minutes for the simcard
thisweekfly: this week flying minutes for the simcard
thismonthfly: this month flying minutes for the simcard

canfly: 10=never (manual), 20=unspecified, 30=no, 40 can accept, 50=can, 60=should, 70=must, 80= always (manual)
importance: sim importance calculated for the next period
flytoorder: engine can accept flying simcard (engines with not so important simcards goes first)

Fly Directions: (tb_flydirections)
defines the fly routes and the minimum times between two gateways
simpacket: applied only for this simpacket (if -1 than applies to all simpacket)
fromgw: fly from this gateway
togw: can fly to this gateway
mintime: minimum time in minutes between the two gateway

Technical Details:

gateway inifile:

```
[phoneX]
flylastupdate=datetime
flyvalidfor= 60000 //seconds
flyto=ip/engine/simpos/simid //template
flyto=local/local/2/999999  //local simpos
flyto=local/2/2/99999999 //other engine
flyto=1.1.1.1/2/3/99999999 //other gateway
flyto=manual  //after manual timetable
flyto=off //switch off
```

tb_sims
origgatewayid: original gateway for the simcard
origline: original engine for the simcard
origsimpos: original position for the simcard
presenterrors: simcard activation fail count (increased in every 10 min if sim is not active when should be active)
preserrreset: presenterrors will be reseted several times to retry the simcard
flyetosim (e): the engine must be bound to the required simcard ?
currflydeep: gateway count from the original gateway
flystarted: time of flying start

staystarted: time of stay start
todayfly: elapsed flying minutes today
thisweekfly: elapsed flying minutes in the current week
thismonthfly: elapsed flying minutes in the current month

4.12.10. GSM Gateways

Deprecated!
Skip this section if you are not using VoIP-GSM gateways.

All configurations can be done from the MManage Client Utility GUI and the VnetCfg utility.
For better understandings we present the gateway configuration settings here:

Phone Settings

```
[PhoneX]
//serial port
PortNumber=1
//control port (not used in 1.6 hardware)
ModemControllPort=X
//if there are no "In" and "Out" device, we use this settings both for in and out
##AudioDevice="Xaaaaaa"
//from engine
AudioDeviceIn="1Audio Codec 1000"
//to engine
AudioDeviceOut="2Audio Codec 1000"
//simchange settings
simchange1= 00:00:00 - 00:00:00 - 01234567890123456789
//if 1 then the conversations (voice) will be saved to files on encrypted, compressed format
record=0
//init commands only for this engine: atinit1,atinit2 ... atinit19
##atinit1=XXXX
##atinit2=XXXX
##etc
//simcard id's in the slots
simcard0=01234567890123456789
simcard1=
simcard2=
simcard3=
```

Simchange settings explanation:

format:

```
  simchange1= 2004.03.05/13:00:00 - 2004.03.07/13:00:00 - 8936302403070132426 (from date - to date)
    or
  simchange2= 10:20:00 - 10:26:00 - 8936302403070132426 (every day from time to time)
    or
  simchange3= 2/10:20:00 - 7/10:26:00 - 8936302403070132426 (from Tuesday 10:00 to  Sunday 10:00)
    or
  simchange4= 6/00:00:00 - 7/24:00:00 - 8936302403070132426 (Saturday and Sunday)
```

there is a priority order from top to bottom (simchange1, simchange2, etc.) numbering begins from 1 without holes
tip: you can set date-hour prioritization
tip: 24:60 is a wrong time (minutes ends with 59)
tip: on day and exact date settings the roundrobin trick is not working
special characters are: - , / . :

Gateway Basic Settings

```
//the name of the gateway. uppercase with "GW" suffix. must be descriptive
alias=NEWGW
//hardware version: 10,16,18 or 19
hwversion=18
//mode of operation. virtual available from hw 1.9
virtualmode=0
//server ip address
serverip=1.2.3.4
//number of hardware audio buffers (the jitter base is soundbuffcount*10)
sndbuffnum=8
//min jittertime in milisec (the minimum of the dynamic maximum jitter time. must be larger than soundbuffcount*5)
minjitter=130
//maximum jittertime in milisec (the maximum of the dynamic maximum jitter time. must be larger than maxjitter. if equal, then static jitter will be applied)
maxjitter=350
//0=off,1=dynamic,2=fixed,3=dynamic+off
silencedetection=3
//codecs to use: onlyg723, onlyg729, onlyg72X, onlyg711
```

```
onlyg72x=1
//useserver if false, then don't connect to the simserver. will save cdr records to file. may be limited due to licensing options
useserver=true
//load configuration from the server (at startup, at regular intervals and when specified)
loadcfgfromdb=true
//gatekkep ip address (leave it empty if you don't want arq registration)
gkip=
//gatekeeper H.235 security
gkpassword=
```

Gateway Advanced Settings

```
//search for gatekeeper
gkdiscover=0
//gatekeeper supported prefixes (from 1 to 100)
gkprefixes1=
gkprefixes2=
gkprefixesX=
//volume in (sound device recorder from the gsm engine). defaults to 40 in hw. 1.8, 100 in hw 1.6
volumein=
//volume out (sound device player to the gsm engine) defaults to 75 in hw. 1.8, 100 in hw 1.6
volumeout=
//gsm engine receive gain. defaults to 0 in hw. 1.8, 64 in hw 1.6
vgr=
//gsm engine transmit gain.  defaults to 0 in hw. 1.8, 64 in hw 1.6
vgt=
//ethernet interface to use. leave it empty to listen on all
netinterface=
//don't touch it usually
launchcmd=voipgsmgw
//install status: 0=idle, 1=wait, 2=normal
opmode=1
//will be set to false after first init
firstinit=true
//tracelevel 1-6 't'
trace=t
//record voice
record=0
```

```
//what kind of logs to send to server (1-5)
tracetoserver=1
//process priority
priority=1
//ModemControllPort used only with hw 1.0
controlportnumber=1
//if we use prefXXX settings
useseparatesettings=0
//signaling endpoint port. Defaults to 20001
mintcpport=20001
//max h323 signaling endpoint port. Defaults to 29999
maxtcpport=29999
//min h323 udp endpoint port. Defaults to 36000
minudpport=36000
//max h323 udp  endpoint port. Defaults to 37999
maxudpport=63999
//min media port. Defaults to 38000
minrtpport=36000
//max media  endpoint port. Defaults to 63999
maxrtpport=63999
//call with immediately pick up
fakecalls=0
//set to 1 if you want error report
errreport=0
//codec frames in one packet: g723frames, g729frames, g72xframes, g72xframes
g72xframes=1
g723frames=1
g729frames=2
//minimum frame count in 1 packet (apply even if the other end says another settings)
g72xminframes=0
//if set to 0, then we send connect when the call arrives
waitforring=1
//reset the engine/gw if we reach this limit
maxnotconnectedcalls=25
//reset the engine/gw if we reach this limit
maxwrongcalls=40
//wrong call criteria
```

```
wrongcallmaxduration=30
//call duration limit in sec (defaults to 3 hour -10800 sec)
callimit=10800
//max time to wait for ring signal from gsm network in msec
maxringewait=36000
//ring limit in msec (defaults to 52 sec)
maxringtime=52000
//deprecated
statusintervall=600
//do Q931 progress indication
doprogreessindicator=0
//reset the gw if we have fewer lines
minactivelines=2
//delay of initialization of the lines (msec)
initdelay=2200
//delay of registration of the lines (msec)
destroydelay=100
//max simchange wait in sec (if sim in call, we will wait until disconnect). default is 5 min
simchangewait=300
//max simcard/channel (will auto detect. don't overwrite)
maxsimcount=
//additional hang-up on the call end (to increase the real duration)
delayonhangup=0
//if we can retry the call
allowreroute=1
//deprecated, as we use only self reroute now
onlyselfreroute=1
//all calls will be routed on the onlyphone if enabled (no simcard requested from the server). deprecated
##onlyphone=3
//automatically increased on every gw (re)start
restartcounter=0
//usually set to 1
enableh245tuneling=1
//usually set to 0
connectwithmedia=1
//usually set to 1
faststart=1
```

```
//used for debug purposes
ringtime=6000
//desktop access
desktoppwd=
//if set, then will try to autologin
loginpwd=
//if we want to play a background sound
backgroundsound=0
//4 or 8. no problem if we use 8 on a 4 channel gateway
chanellnum=8
//pincode applied globally to all channels (if not specified in phonex section)
pincode=
//will set the simcards to don't request for pincode (pincode must be set in gateway or phonex sections)
autoremovepincode=true
//volume in/out (will be overwritten with volumein and volumeout)
volume=
//auto gain enable/disable
doautogain=0
//listening tcp port (may be changed on NAT configurations)
signalport=1721
//0=no watchdog, 1=yes, 2=unknown
paralellwatchdog=2
//set to 1 if you want to remap usb audio lines
mustremapaudio=0
//set to 1 if you want to reread all simcards
readallsims=0
//set to 0 if you don't want an usb remap on every pc restart
canremaponstart=1
//if we have usb audio and don’t have other usb device then allow to remap if needed
canremapusbaudio=1
canrenewusbaudio=1
//set to 0 if you don’t want panel reset
canpanelreset=1
//set to 0 if you want an usb remap when the service will start
mustremapaudio=0
//disable reading sms messages
nosmsread=0
```

```
//socket read/write timeout and system checks operations modifier. default=4
timeoutmultiplier=4
//backup server address
serverip2=
//route incoming calls here (defaults to serverip if not specified)
outserverip=
//keep connected to the internet (redial, reconnect, repair, enable/disable network interface, restart)
keepinternet=1
//ethernet interface name. configure from the vnetcfg tool
net_interfacename=
//network connection type (STATICIP/DHCPIP/ISDNIP/ADSLIP,CARDNAME). configure from the vnetcfg tool
net_conntype=
//network interface ip address. configure from the vnetcfg tool
net_ip=
//network netmask. configure from the vnetcfg tool
net_netmask=
//network default gateway. configure from the vnetcfg tool
net_defgw=
//network primary dns server. configure from the vnetcfg tool
net_dns=
//dialup phone number
net_phonenum=
//network ppp username. configure from the vnetcfg tool
net_username=
//network ppp password. configure from the vnetcfg tool
net_pwd=
//maximum speech length allowed in sec. defaults to 10800 (3 hour). set to 0 to disable
maxcallduration=
//maximum ringtime allowed in msec. defaults to 52000 (52 sec)
maxringtime=
//password on local command line. default is cmdpwd1234
cmdpwd=cmdpwd1234
//towarding dtmf from voip to gsm
forwarddtmf=1
//what to do with incoming calls (0=drop,1=hold a little then drop,2=auto forward,3=forward to server as forwardnum,4=forward to number requested by
dtmf)
inccalls=1
```

```
//file to play when requesting number to call on dtmf (when incalls is 4). "please enter phone number to forward call"
playdtmfreqfile=
//file to play when requesting number to call on dtmf failed (when incalls is 4) "forwarding failed"
playdtmffail=
//file to play when requesting number to call on dtmf succeed, and forwarding begins (when incalls is 4) "your call has been forwarded. please wait for connect"
playdtmfforward=
//auto forward number (used if inccalls is 2)
forwardnum=
//used to require the number to forward to (when inccalls is 4)
promtfile=
//allow towarding dtmf messages to gsm network
allowdtmf=true
//how we send the ring signal. 0=send immediately and always, 1=send when received from gsm, (on the server you can set a timeout)
exactring=1
//used by the ipconfig tool. don't edit manually
ethcfg=
//local ip stored here. don't modify
localip=
//date-time of the last config download from the server
lastinisave=
//date-time of the last config upload to the server
lastiniupload=
```

Watchdog settings

```
 [watchdog]
//set to 0 if you don’t want pc restarts
canrestartpc=1
//set to 0 if you don’t want service restart (then the watchdog will have no effect)
canrestartservice=1
//set to 1 if you want a reset on every night
canrestartdaily=0
//how often can the watchdog restart the service. defaults to 1000*60*25 msec (will change dynamically)
MAXSERVICERESTARTIVAL=
//how often can the watchdog restart the pc. defaults to 1000*60*45 msec (will change dynamically)
MAXPCRESTARTIVAL=
//max time to wait for watchdog reset. defaults to 1000*60*20 msec
```

```
MUSTRECEIVEOKIVAL=
```

Other settings

```
//at commands sent only once for all engines
[atonce]
#hardware version
cmd0=AT+WHWV
#sw version
cmd1=AT+WSSW

//at commands sent for all engine at every init
[atinit]
##cmd0=XXXX
##cmd1=XXXX
##etc

//prefix depending settings
[prefXXX]
connectwithmedia=0
g723frames=3
g729frames=6

[ipmux]
ipmuxenabled=0/1

[sounddevices]
//will be filled when reading all sims, so you can copy device names from here
```

Handling incoming calls from GSM network

Depending on the “incalls” (gateway configuration) settings, incoming calls from gsm network can be handled in several ways.

1. When incalls is set to 0

-all incoming calls to gsm simcards will be dropped immediately

2. When incalls is set to 1

-the engine will pickup the call, hold a little (random time, but maximum 1 minute), and than drop. Also used in call simulations.

3. When incalls is set to 2

-call will be forwarded to the number specified by the “forwardnum“ option in the GSM network.

-the simcards must support the forwarding options, otherwise this operation will fail

4. When incalls is set to 3

-the call will be forwarded to the Mizu server specified by the “outserverip” setting in the gateway configuration.
-on the server, the call will be forwarded to the “gsminccalled” number (SimPlatform configuration). If the “gsminccaller” option is filled with a valid phone number, than the callernumber will change accordingly. Otherwise the caller number will be the original caller. The ip caller address can be changed with the “gsminccallerip” option. (thus you can simulate the routing from a predefined user)

5. When incalls is set to 4

-the caller will be asked to enter the target number (handled with dtmf), and the call will be forwarded to that number
-the prompt played to ask the target number can be set by the “playdtmfreqfile” setting. This will have to point to a PCM 8000kHz, 8 bit mono wave audio file.
-the prompt to be played if the forwarding has failed can be specified by the “playdtmffail” setting. When the forwarding is in progress, the “playdtmfforward” file will be played to the user.
-the call will arrive to the server with the ‘222’ techprefix, and you can setup a separate routing roule for this tecprefix

A possibe callflow for calls arrived to a working simcard in” inccals 3” mode:

```
original caller initiate call
  call routed by gsm network
    call arrives to our simcard
      converting from gsm to h323
        call routed to GK
          on routing: caller will be converted to “gsminccaller”, called will be converted to “gsminccalled”
              call needs to be forwarded to “sip2h323”
                   sip2h323 module doing the protocoll conversion from h323 to sip
                      call arrives to SIP module
                         routing will decide the proper routing
                            client sip endpoint is created and call forwarded to the target gateway
                               call will be handled by the called gateway
                                  call forwarded to the final user
```

Operator friendly gsm termination

Not using industrial engines

On request, we can deploy our gateways equipped with normal gsm phones instead of industrial gsm engines. Ask the Mizu support for more details

Virtual Engines

Each simcards can have it’s own GSM engine (in other gsm gateway the engines are used by more simcards)

GSM Cell Lock

Because Mizu GSM Gateways use only 8 channels, they don’t overuse the gsm network. However, you can setup individual GSM channels to use separate cells

Virtual Simcards
With the ease of Mizu simbank, your simcards can be stored in a central location, and used in gsm gateways installed at different locations.
Delayed network registration
A delay time can be configured to elapse between succesive engine (re)registrations.
Intelligent routing
Ballancing the traffic across your simcard based on price and quality
Handling of incoming calls
In usual GSM gateways there are no simple mechanisms to handle incoming calls from the gsm network. In a Mizu system all calls can be forwarded to your support team, so each call can be responded accordingly.
No GSM network owerload
Mizu GSM gateways occupy only 8 channels
Fast detection of dead channels
Failovering from simcards blocked by the operator or with wrong quality
Automatic blacklist generation
Wrong numbers will be detected and blocked on the server (not forwarded to the gsm network)
Minute limits
Each simcard can have different daily, monthly and other limits
Time between subsequent calls
Calls will not be forwarded to gsm gateway without a delay between (configurable)
SIM Bank
Simcards can be managed on a central location, and they will “fly” between gateways
Many other tricks
Ask the Mizu support for more details

How to setup a gateway behind a NAT

For the basic gateway the followin port forward must be applied on your NAT box:
 -TCP: 1721 (h323 signaling)
 -TCP: 3386 (remote desktop)
 -UDP: 44444 (“voice-here”)
 -at least 16 UDP port for RTP (setup the minrtpport and maxrtpport settings accordingly in the voipgsmgw.ini)
  (RTP media may work without the port forwarding set explicitly. This depends on caller party)
 -TCP: 21 (optional for ftp access)

For advanced gateways and servers the following port ports must be forwarded also:
 -TCP: 1720 (default h323 GK signaling)
 -UDP: 5060 (default SIP port)
 -TCP: 9885, 9886, 9889 (gsm server, admin port, log port)

-TCP: 1433 or 2223 (for SQL server)
-TCP: 80 (optional for HTTP)

**you may use other ports than the defaults listed above*

4.12.11. H.323

The mizu voip server is capable to handle H.323-H.323 routing or transparent sip to H.323or H.323 to sip conversion.
Both Gateway and Gatekeeper (v.1,2,3,4) mode are enabled and protocol variations are handled automatically (RAS, H.245, H.225, media proxy or bypass, Fast Connect/Fast Start, etc)
H.323 should be used only for gateways. For endusers you should always prefer SIP over H323 because:
-some class 5 features will work only with SIP protocol
-H323 GK doesn't support username/password authentication (only ip/port and/or techprefix based)

Files needed (part of the install package):

- atarongk.exe -main h323 gateway and gatekeeper
- vsip.exe -for SIP to H323 conversion (will not be used for H323 to H323 calls)
- openh323.dll, ptlib.dll, pwlib.dll, libyate.dll -required dll's
- sipcfg directory and sipunits directory -required for sip to h323 converter (vsip.exe)

To enable h323 the hash323 and can_h323 global config values must be set to true.
To enable h323-SIP conversion the runsipproxy global config values must be set to true.
Local and LAN IP's should be also enabled.
These are can be enabled from the Configuration Wizard.
The atarongk application can be controlled from the console port if you type the "gk" command

Two important **user entry** is created automatically during setup used internally for sip to h323 conversion (and inverse):

- "sip2h323caller" traffic sender
- "sip2h323" sip proxy

Make sue that these are present and enabled.

To setup a h323 caller (**incoming traffic**), there is no special action to do. You just have to add a traffic sender user with the proper authentication.
For **outgoing traffic** you must use a "H323 GW/GK" user type and then add it to your routing

In case of H323 entries, the "**callsigaddress**" field is used instead of the "port" field. This must be set to 1720 usually (standard H323 signaling port). The media ports are negotiated automatically.

Make sure that atarongk and vsip is enabled on the firewall.

Technical background:
h323 to h323:

1. incoming call arrive to atarongk
2. atarongk will call the routing and will get the destination address
3. atarongk forwards the call to the destination

h323 to sip conversion:

1. incoming call arrive to atarongk (h323 protocol)
2. atarongk will call the routing
3. call will be forwarded to vsip converter
4. vsip will call the routing again (sip protocol)
5. the rest of the call is handled by mizu sipstack

sip to h323 conversion:

1. incoming call arrive to mizu sipstack (sip protocol)
2. call will be sent for vsip
3. vsip will send the call to atarongk (h323 protocol)
4. atarongk will call the routing again
5. atarongk will forward the call to the destination

Troubleshooting:

1. replace the atarongk file with the _dbg version
2. change the "sipcommand" global config option to "vsip -vvvvvvvv -l siplog.dat"
3. change the "gkcommand" global config option to "atarongk -ttttt -o gklog.txt"
4. set loglevel to 5
5. check the callsigaddress field for h323 gateways (usually 1720)
6. start the service
7. make sure that atarongk and vsip processes are running
8. make a test call and check the logs
9. for h323 test calls you can use the openphone.exe or the ohphone.exe (example: ohphone -n -p 192.168.1.7 -ttttt 2222)

4.12.12. Encryption and tunneling

The mizu voip service and client application has built in tunneling and encryption features.

For a detailed presentation please visit this link: https://www.mizu-voip.com/Software/VoIPTunnel.aspx
You can configure the most important settings on the "Tunnel" page of the server configuration wizard from MManage Config menu.
The secured communication can used by all softphones provided by mizutech (customized softphones for all platforms with buit-in tunneling and encryption support) or you can use the cross-platform "mtunnelclient" application to secure any third-party server/devices/phones.

Encryption v.2 (new)

The new version of the tunneling module brings many advantages as described on the website. Highly recommended to upgrade if you are using the old version.

basic settings:

- useencryption: 0=disabled, 1=only when rec encrypted (default), 2=use, 3=always (only 0 and 1 is used for now)
- fwdregistrations: forward registrations to upper server and don't handle locally. 0=no,1=only from alternate port, 2=always
- forwardauthentifications: will forward invite (regarding routing setup) 0=no (default),1=yes,2=yes with username as callername, 3=yes with phonenumber as callername, 4=yes with username as authname too, 5=yes with phonenumber as authname too, 6=replace authorization with username but leave the A number intact, 7=replace authorization with sipphonenumber
- other related: replacecalleronforward, forwardauthentificationsfromtr, forwardauthentificationsfromend, replacecalleriponforward, normalizedef,
- autocreatereguser: 0=no,1=when fwd authenticated ok register, 2=always (when we receive the register)
- userautoaddwithowner: wether server should auto create resellers. -1=auto,0=no (standalone. no resellers needed), 1=yes (less secure), 2=must (must means that the calls will be rejected if parent not already exists or no fix match for brandname)
- allowupperserverselection: 0=no,1=yes (def), 2=yes only if found as sipserver   allow client softphone to specify upper server. default is true. for standalone sofswitch with prebuilt softphone should be 0. otherwise can be 1 or 2
- fwdregistrations_domain=fix registrar sip domain
- fwdregistrations_ip=fix registrar ip or FQDNS
- fwdregistrations_port=fix registrar port

other settings:

- maxlinefornewunknownparents: def max line for auto created resellers (default is 5). useful to auto create trial access together if userautoaddwithowner is set to 2
- maxlinefornewunknownusers: max line for new unknown users. (default values is 3)
- alternatelocalportencrypt: 0=default,1=never,2=auto,3=always,4=reject if not encrypted (To encrypt all communication on the alternatelocalport. If set to 3, then will not accept clear test VoIP)
- usequickencryption: setting for v1. false for strong but slower encryption mode , true for weak and quick encryption mode. default is true.

- alloweuserusercalls = 1: 0=no (don't check if local target),1=yes (default),2=disable (dropp) call to endusers route calls directly between endusers or all calls have to go trough upperserver. default is 1.
- encryptedpeerlist: list of IP to always encrypt (optional)
- noencryptedpeerlist: list of IP to never encrypt (optional)
- noencryptedpeerlist2: list of IP to never encrypt on socket send level (optional)
- tb_users.encrypt: 0=default 1=newer 2=dynamic no (decide automatically) 3=dynamic 4=dynamic yes 5=always (to always initiate encrypted sessions) 6=force always
- tb_users.candisablesim: will auto store the encryption method. 0=not known, 1-20 old encryption, 21=new encryption, 22=new fast encryption (for iOS)

Encryption v.1 (old)

Optionally a separate udp port can be set on the server to handle encrypted sessions (alternatelocalport)

Encrypted sessions are always answered encrypted by the server.
Use the tb_users.encrypt field to control encryption on user level:
0=default
1=newer
2=dynamic no
3=dynamic
4=dynamic yes
5=always
6=force always
To initiate encrypted sessions, set the “encrypt” field for the user to 5.
To decide automatically, then set to 2.
To encrypt all communication on the alternatelocalport, set “alternatelocalportencrypt” global config to 2.

There are two types of built-in encryption:
-weak and quick encryption mode (when usequickencryption is true -default)
-strong but slower encryption mode (when usequickencryption is false)

Encryption can be enabled/disabled with the “useencryption” global configuration value:
0=disabled,
1=only when rec encrypted (default)
2=use
3=always
The encryption method is stored in the “candisablesim” field in tb_users (refreshed automatically by the server on client connect)

Also, there is a possibility to define a list of ip addresses with the “encryptedpeerlist”. All communications with these peers will be done encrypted.

If the mvoiptunnel client service is used with some always running device, then you should set a fix “encryptionkey”, otherwise it is changed every night which might cause communication issues for long running clients.

Standard SSL/TLS signaling encryptions are negotiated runtime as well as SRTP media encryption.
The recommended encryption type between mizu devices is the embedded fast or bowlish+compression. These types of encryption require much less CPU power and are done with no additional network overhead.

Running Mizu Server in proxy mode

(the server doesn’t handle registration and authentication, just forward it)

```
Fwdregistrations=2 // //0=no,1=only from alternate port, 2=always
fwdregistrations_domain=registrar sip domain
fwdregistrations_ip=registrar ip or FQDNS
fwdregistrations_port=registrar port
autocreatereguser=1 //0=no,1=when fwd authenticated ok register, 2=always (when we receive the register)
Forwardauthentifications=1 //will forward invite (regarding routing setup)
Sendrandompackets=true
udptunnelsamesock=1
Ftpname=-
Allowdiscmessage=false
Maxreroute=1 //only if 1 upper server is used
Sslogport=0
```

Setup routing to point to the upper server

Optionaly you may increase these tresholds:

```
        Maxmsgcountlimitmultiplier=2
        Maxnocdrmin=3000
        Maxroutereqpermin=5000
        Checkmaxlines = 0 ?
        Checkmaxlinetb=0 ?
        Maxsessionspeechlen= 1000UL * 60UL * 60UL * 6UL
        Ringtimeout=120
        MAXEPCOUNTTRESHOLD=30000
        MAXSUBSMSGCOUNT=99999
        MAXWRONGMSGALLOWED=9999
```

Running the server as a VPN access point (forced encryption)

```
Fwdregistrations=1 //only from encrypted clients (bug: set to 2)
```

Alternatelocalport=5088 //or any other port; same like the port that was set in the softphone and webphone
Alternatelocalportencrypt=3 //0=default,1=never,2=auto,3=always
normalize_clean=1
normalizenumbers=0
alloweuserusercalls=false //set to disable user to user calls
Set the “encrypt” field for the users to 5 (update tb_users set encrypt = 5 where type = 0)

When all the new automatically created users are allowed to use only encrypted comunication, set the autonewusersencrypt to 5.

Stop and disable the MizuWebService

TCP and HTTP tunneling port settings
-on the server, set the usehttp and usetcptunnel settings to true
-the default port for simple TCP signaling is the same as the main udp port (5060 by default and configurable with the “localtcpport” parameter)
-the default port for http tunneling is 80. Set it explicitely if needed with the localhttpport parameter. In the softphone it is controlled by the remotehttpport parameter
-the default port for tcp tunneling is 443. Set it explicitely if needed with the localtcptunnelport parameter. In the softphone it is controlled by the remotetcptunnelingport parameter
-the default port for TLS signaling is the same as the main udp port + 1 (5061 by default)

Client driven upper server
Setup the tunneling server as usually (create a “default” registrar and upper server)
Set “allowupperserverselection” to “true”.
Create a sip proxy named “upperserver_template”. (this will be a “default” route and default upper server settings)
Create one or more server and set the “rdport” field properly.
From the client set the sy.uppersrv properly (must match with a rdport or set to a upper server ip:port)
This can be done with the upperserver applet parameter or with the UPPERSERVERID define from the softphones.

Tunneling signaling and media in the same udp stream
Set the udptunnelsamesock congig option to at least 1. (udptunnelsamesock = 0; //0=no,1=if received so, 2=yes). This will tunnel only encrypted sessions so it is safe to use.
Increase the “udpbuffsize” configuration option

Tunneling between two servers
Setup the sip servers and traffic senders with forced encryption and desired protocol
Set the “usehttpclient“ global config option to “true”.
Set tb_users.protocol to 4 if TCP tunneling is needed and set the port to 443 (default TCP server port)

Tunneling between a servers and mtunnelsrv
Just set the tunnel client to connect to the server.
You might also adjust the following tcp tunneling related configuration options:
-tcptunnelallowsendtosameip; //0=no,1=without streamid,2=yes
-tcptunnelmultipleep; //when set to true than we are searching for rtp ssrc (more than one ep can send in one tcp stream. for example when we have a mtunnelclientsrv with multiple gateways)
-userptepptrsig; //each tcp rtp packet is marked when this is set to true

4.12.13. SBC

The mizu voip service can be used as an SBC, providing:

- transparent user authentication (will forward digest authentications to your PBX or softswitch)
- SIP security at the edge of your network
- WebRTC-SIP gateway functionality
- H.323-SIP gateway functionality
- rate limiting
- NAT traversal support
- media transcoding
- call and chat recording
- fast route between local endpoints to free up server resources
- and other services

You can turn the mizu server to an SBC from the configuration wizard or via the following global config options:

- autocreatereguser=1 //0=no,1=when fwd authenticated ok register, 2=always (when we receive the register)
- fwdregistrations=2 //0=no,1=only from alternate port, 2=always
- fwdregistrations_address/fwdregistrations_domain/fwdregistrations_ip/fwdregistrations_port
- forwardauthentifications=1 //0=no (default),1=yes,2=yes with username as callername, 3=yes with phonenumber as callername, 4=yes with username as authname too, 5=yes with phonenumber as authname too, 6=replace authorization with username but leave the A number intact, 7=replace authorization with sipphonenumber
- forwardauthpassword=1 //-1=auto, 0=fwd from ep, 1=means 3 if from webrtc for recent users, 2=answer from local always,3=answer from localalso for register, 4=answer from remapped upperusername/upperpassword
- fwdregistrations_xxx settings: fwdregistrations_domain, fwdregistrations_address, fwdregistrations_ip, fwdregistrations_port, fwdregistrations_protocol, fwdregistrations_proxy_ip, fwdregistrations_proxy_port

When running as an SBC the registrations and calls will be forwarded to:

- the globally configured upper server (fwdregistrations_xxx)
- or after routing rules (Always for the calls. For registrations you need to set the routingforregister setting to 1 for this)
- or as suggested by the SIP client UA (if enabled by setting the allowupperserverselection setting to 1)

A more details SBC guide can be found here:
https://www.mizu-voip.com/Portals/0/Files/SIP_SBC_Doc.pdf

4.13. Security and account limiting

The Mizu VoIP server is secure by default however with also usability in mind. For usual usage you can leave everything with the default settings and you still have a high level protection as the Mizu server has security constrains and built-in automatic attack prevention mechanisms at multiple levels enabled by default. The Mizu VoIP server has very few dependencies of third-party components (including OS services) which also improves security.
Please note that regardless of the server side security, there are still a room for hackers to exploit client endpoints such as stooling SIP username/passwords from softphones/devices. You can't prevent these to happen as it depends on client environments and third party software/hardware however you can still reduce its impact by leveraging strength server side restrictions (max lines, max daily/monthly limits and others).
For extra security, read through this chapter and fine-tune your settings, however please note that most of these settings will have direct effect on usability. (Your service might work incorrectly if security restrictions are too strength. Don’t change setting unnecessarily)

4.13.1. OS security

The Mizu VoIP server uses only core windows services (networking, file access and other resources with no security implications) so it is not vulnerable for the usual windows related vulnerabilities, because kernel/core ip networking vulnerabilities are very rare and hard to exploit (and windows is no worse in this then other systems such as like linux).
For maximum security just use a clean windows install, with no any unneeded services enabled and see the OS security checklist at the end of this chapter.
You can configure your MizuVoIP server to disable all network services on your firewall (including RDP and SQL access) and enable only the VoIP service and the mztunnel (you can still comfortably manage all services from remote MManage via the mztunnel).

4.13.2. DB security

The database contains ALL the data used by the Mizu server including user details and CDR records so a special care has to be taken to protect your database.
With a regular SQL install using a strong password you are already secure by default without any special action to be taken.
SQL injection attacks are prevented on multiple layers: all input data are properly filtered and sanitized. SQL parameters are passed by query parameters and not with SQL query string manipulation using stored procedures whenever practicable.
Note: the MManage admin client is treated as a trusted component with few restrictions once the user is logged in.
Make sure to have working backups (incremental/full) to be able to recover your database from a disaster at any time.

4.13.3. Socket/stream level network protection

All streams are rate limited in the server to avoid all DoS or overflow attempts.
To fine-tune, you might change the "maxnetworkspeed" global config option which is set to 800000 by default (this means 0.8 gbits so it is optimized for a gigabit NIC with some room for IP/TCP/UDP headers)
For example if your NIC is only 100 mbits, then you might set this to 80000.
The server will perform early message inspection at network level to efficiently filter out malicious requests. These and other extra protections are also applied for media streams for RTP/RTCP flood protection.

4.13.4. Address level network attack preventions

- DOS attack prevention: when there are too many messages received from an IP address, the address will be blacklisted automatically. Controlled by MAXSUBSMSGCOUNT (default is 39000) and MAXSUBSMSGPERIOD (default is 3 minute) global configuration settings.
- when there are too many "wrong" or meaningless messages from an IP, the address will be blacklisted automatically. Is controlled by MAXWRONGMSGALLOWED (default is 4000) and MAXWRONGMSGPERIOD (default is 1 hour) global configuration setting.
- when there are too many subsequent auth request from an IP with no positive result, the source address will be blacklisted automatically. Is controlled by maxsubsipauthrequest global configuration setting (default is -1 which means automatically calculated based on server load)
- when there are too many unauthorized request from an IP, the address will be blacklisted automatically. Is controlled by MAXWRONGAUTHFROMIP (default is 2500) and MAXSUBFAILAUTHPERIOD (default is 1 hour) global configuration setting.
- when there are too many unauthorized request from an IP:port, the address will be blacklisted automatically. Is controlled by MAXFAILEDAUTHENABLEDIPPORT (default is 50)/ MAXFAILEDAUTHENABLEDIP (default is 800) and MAXSUBFAILAUTHPERIOD (default is 1 hour) global configuration setting.
- You might set the "checkrtpaddr" configuration option to 2 to force the server to accept rtp packets only from known media sources. (the default setting is false)
- All counters are reset periodically controlled by the REENABLEDOSBLOCKED (default is 6 hours) global configuration setting.
- For the new registrar module the "fastauthsubswrongusermsg" setting will be applied: not authenticated messages from same ip or all messages in 90 sec. (also X*10 for any IP). Default value is 300.

4.13.5. Session level protection

Session level limits are enabled for all services provided by the Mizu VoIP server. This includes the core VoIP services (SIP/H323) but also the enduser web portal and the API.
The server will close the session on the following circumstances:

- absolute timeout, call timeout, media timeout, ring timeout, call init timeout, timeout on session timers (all these are configurable from MManage -> Configurations form)
- too many incoming messages (dynamically based on frequency and number of requests)
- too many authentication failures (dynamically based on frequency and number of requests)
- too quick (abnormal) message receptions (dynamically based on frequency and number of requests)

- subsequent invalid signaling messages (DDOS protection)
- some known scanners are blocked by default. You can add more with the "**blockua**" global config option

Blocked devices and users can be re-enabled by issuing the “delbanned,ip” or “delbanned,all” command on the Console port.

4.13.6. Web security

The Mizu VoIP server uses a native embedded engine for web interfaces so it is not vulnerable for the exploits against the common web servers.
The web engine implements the best current practices including proper authentication, input filtering, rate limiting and XSS attack prevention with secured default settings.

4.13.7. API access security

The API is secured by default and each sensitive API requires proper authentication to be executed. All data received from the network or user input are properly filtered and sanitized.
Optionally you can harden its security by setting up HTTPS access, setting a random httpapikey key, fine-tuning the API access and using the methods described in the API documentation.
See the "Security", the "Authentication" and the “Hardened access” sections in the VoIP_Server_API documentation.

4.13.8. Payment security

User payments security are completely offloaded to payment gateways (for example PayPal), which must comply for PCI requirements.
This means that the Mizu Server doesn't process any confidential data such as user or credit card details, just forwards the user to the configured payment processor.
Once the transaction is completed (or failed) the Mizu server will receive a message from the payment processor to handle the result (increase user credit).
Various frauds are not uncommon with online payments which are handled by both the payment processor and the mizu server. Suspicious transaction can be left to be approved manually instead to be accepted automatically.
Check your PayPal related settings to enforce proper verifications and limits. (See the PayPal related section in this documentation and search for “paypal” in the MManage configuration form)

4.13.9. Encrypted VoIP

The Mizu VoIP server provides a complete solution for secured communication. See the "Encryption and tunneling" section and the VoIP tunnel documentation for the details.

4.13.10. SSL/TLS/WSS/HTTPS setup

You can use the TLS proxy module to add secured transport for most protocols in the mizu server. This module can be used to encrypt all services offered by the mizu voip server, automatically handling each built-in protocols separately regarding to its requirements, including:

- SIPS (secure SIP protocol. Default port is 5061. Unencrypted port: 5060)

- VoIP tunneling (tunneling over TLS. Default port is random)
- WebRTC (secure websocket / wss)
- HTTPS (Secure HTTP protocol. Default port is 443. Unencrypted port: 80). This is used for the following services:
    - enduser web portal (default internal unencrypted port is 8080)
    - API (default internal unencrypted port is 80)
    - mmq service (default internal unencrypted port is 80)
    - websocket (for API, WebRTC and mmq)
    - the universal port 80 (unified maina port)

The VoIP server is capable to maintain a Let's Encrypt TLS certificate automatically which can be selected from the configuration wizard.
You will need to assign a domain name to your server to be able to get a TLS certificate. Create a (sub)domain at your domain service provide (for example sip.yourcompany.com) and set its DNS A and/or SRV record to point to your server IP address.
For other related settings, search for "tls" and "ssl" in the global config (Configuration form).

Note: From v.7.5.2 the mizu server is capable to automatically acquire SSL certificate from Let's Encrypt and it has built-in TLS module (no need for the TLS Proxy anymore), so there is no need for be below described steps anymore.
By default the server is shipped with a self-signed certificate. This can be used for VoIP tunnel encryption, but not for authentication/identity. Will not work for wss and if used for the webportal, the browsers will present a warning.
Follow these steps to manually setup and enable TLS on your server with a valid certificate (less likely required for new versions):

1. You need a (sub)domain name assigned to your server IP (you should also set it in "LocalDomain" global config).
   If you already have a domain name (such as company.com/www.company.com) then with most of the DNS providers you can create sub-domains for free from their control panel. Create a subdomain like sip.company.com for your VoIP server IP. If you don't have a domain name yet, then you can purchase for any DNS providers at around $10/year. For example at GoDaddy.
   You can use the same subdomain for both your web and voip server is you wish.
2. Get a valid certificate for your domain from a reputable CA
   StartSSL: free, but you can't use it to host the webhone.jar (Not recognized by Oracle, however good for WebRTC or any other purposes)
   Comodo: $10 per year. Works everywhere
   Let's Encrypt: free (recommended for sysadmins and developers).
   For example you can get free certificate from StartSSL by following these steps or for let's encrypt here is an easy app.
   You can use the same certificate for both your webserver and VoIP server if they are on the same IP with the same (sub)domain.
3. Make sure the tlsproxy.exe, tlsproxy_mserver.exe or tlsproxy_servicename.exe exists in your app directory (contact us if you can't find this app in your server directory)
4. Add your SSL certificate. For this you just have to copy the following files in the server directory:
    - key.pem: the key file (your private key which you have got when generated the CSR - keyfile.key)
    - cert.pem: the certificate file (your certificate as received from the CA -yourdomaincertificate)
    - root.pem (the chain: CA intermediary + root certificate. intermediate.crt + root.crt)
    - dh.pem and/or dh2.pem: these are optional used for DH key exchange and it can be generated with the following commands:
      -fast: openssl dhparam -dsaparam -out dh.pem 1048
      -slower/better: openssl dhparam -out dh2.pem 2048

   All these are clear text files. If you haven't received your certificate in similar format from your CA, then convert them and create these files (copy content using notepad). If these files are not set, then the server will use the default built-in self-signed certificate (which will result in browser warnings).

5. Set server settings (search for “tls” in the global configuration)
    - usetls: set to true for SIPS (SIP signaling encryption)
    - encv2_usetls: set to true if you wish to use TLS for VoIP tunneling (if you are using the VoIP tunneling module)
    - usetlsforweb: for internal webportal
    - usetlsforall: will auto set TLS for mainaport, web, voip tunnel, sips (recommended)
    - forcetls: runtime http -> https rewrite: 0=force no, 1=def, 2=force tls,3=extra
    - localtlsport: SIPS TLS port (default is 5061)
    - sslcertpwd: if your certificate is password encypted, set the password here
    - set the domain name assigned to your VoIP server to the “localdomain” global config
    - you might also need set the cfg_baseurl webportal setting to https:\\yourdomain
    - if you need to use an other port then the standard 443, then set it in the sslportweb config (also it can be set in the tlsproxy.ini)
6. Configure tlsproxy.ini if you need the certificate for WebRTC/websocket. Example:

    ```
    [settings]
    security=1
    insertheader=1
    forwardtoothers=1
    localdomain=sub.domain.com
    externalip=11.22.33.44
    listenip=192.168.1.8
    listenport=2443
    ```
7. Reload or restart the service (restart recommended)
8. Test:

    - Direct response from the proxy: https://yoursipdomain.com/ tlsproxytestxxxxxxx (only for gateways if forwardtoothers is set)
    - Example request for enduser webportal over the unified port: https://yoursipdomain.com/webvoip
    - Or via a request to built-in webportal: https://yoursipdomain.com/mvweb/anyexistingfile.html
    - For api: https://yoursipdomain.com/mvapireq/?apientry=apitest1

    (Note: make sure to add the :port after the domain if your tls is not on the standard 443 port)

Other tools:

- SSLBuddy
- OpenSSL (win binaries)
- Test

4.13.11. User authentication

The mizu server provides flexible user authentication services which can be set per user by the "AuthType" config option from MManage. See the "Users" and the "User authorization" section for more details. On the IVR the users can be authenticated also by PIN code or A number authentication. See the “IVR” section or the IVR documentation for the details.

For speed considerations, the mizu server can cache device login information for a time so it will not ask again for authorization for every REGISTER or INVITE request. This means that it can happen that you change the user credentials and the user is still able to login with its old username/password or inverse: the user enters the correct credential but will be still blocked for a time. This can be controlled by the "cacheregistrations" global config variable.
The server will periodically recheck the password for all accounts and will change to a random strong password when weak password is found (enforcestrongauth global config option). Password storage can be encrypted/hashed as set by the securepasswords global config option.

4.13.12. Per IP call limits

You can restrict the maximum number of calls or call duration from a source address by the following global settings:

- maxcallperip: max number of calls from the same IP
- maxdurationperip: max call duration from the same IP
- maxcostperip: max call cost per IP
- maxcallperperiod: the period for the above (1.0 means one day)
- maxcallperuserid: the above will be set to this user only (set to -1 for all users)

Additionally to IP restriction you should set as much other restrictions as possible to restrict your server attack surface, such restricting the usage to one user (maxcallperuserid), set user as prepaid, set max lines and daily/monthly limits on the account as other best practices discussed in this documentation.

4.13.13. IP Spoofing

IP Spoofing means incoming TCP or UDP packets with false source IP which might be used by attackers to impersonalization. You can avoid this with the following measurements:

- If you are vulnerable to IP spoofing, make sure to not use IP based authentication.
- Consider using a techprefix also of IP authentication is required for call (NeedAuth user field set to 3 which means IP auth + tech prefix)
- Make sure that your router or gateway performs proper ingress filtering (blocking of packets from outside the network with a source address inside the network).
- Adjust the allowtrustedip global config option after your needs: 0=no,1=my ip only,2=private ip's only,3=yes(def),4=everywhere(including admin console)

4.13.14. Firewall

The server has a built-in static and dynamic firewall to block unwanted sources. See the "Firewall" section in this documentation for more details.

4.13.15. Maximum simultaneous call limits

To avoid overflow with calls you should set the MaxLines/MaxLinesOffpeak settings for each enduser and traffic sender properly.
(For enduser the default Max Line is 5. We don't recommend to be set to 1, since this might disallow services such as call transfer and conference)

4.13.16. Prepaid account credit limits

Prepaid accounts have a hard limit by their credit value.

4.13.17. Postpaid account monthly spend limits

You can also limit postpaid account to prevent unexpected bills.

Set the “creditcheckforpostp” global config to true.
Set a meaningful default and actual value for each user for the followings: maxmonthlycredit, maxmonthlycreditinc, maxmonthlycreditend.

- maxmonthlycredit: max allowed credit/month even if the user is postpaid
- maxmonthlycreditend: max Maxmonthlycredit (because we increase Maxmonthlycredit by maxmonthlycreditinc every month if the user was active)
- onlylocalaccess: traffic sender traffic will not be forwarded (can call only local users from tb_users and tb_numbers)
- maxmonthlycreditinc: determines how much money we add to Maxmonthlycredit every month

To set the default value, you should edit the database default values and then run a query like this:

```
update tb_users set maxmonthlycredit = X, maxmonthlycreditinc=X, maxmonthlycreditend=X where type = 0 and postpaid > 0
```

To make it default for new users:

```
alter table tb_users drop constraint DF_tb_users_maxmonthlycredit
alter table tb_users drop constraint DF_tb_users_maxmonthlycreditend
alter table tb_users drop constraint DF_tb_users_maxmonthlycreditinc

alter table tb_users add constraint DF_tb_users_maxmonthlycredit default X for maxmonthlycredit
alter table tb_users add constraint DF_tb_users_maxmonthlycreditend default X for maxmonthlycreditend
alter table tb_users add constraint DF_tb_users_maxmonthlycreditinc default X for maxmonthlycreditinc
```

4.13.18. Fix daily credit limits

You can apply a maximum daily limit for all (both prepaid and postpaid) accounts with the following global configurations settings:

- maxdailycreditforenduser (max daily spend for endusers)
- maxdailycreditforts (max daily spend for traffic senders)
- You can also set the limit on user level by setting the "maxdailycredit" field for the respective users. (For this to work you should also set the "maxdailycreditperuser" global config option to 1, but that is auto set at first config reload if at least one user has "maxdailycredit" set)

4.13.19. Dynamic credit/spent limits

Set the "checkmaxdyndailycredit" global configuration option to enable dynamic credit/spent monitoring and call block on threshold.
Once this is turned on, the server will not allow the usage (spending) to grow more than a specified level (maxdyndailycredit_multiplier) within a single day. (The server will always remember the maximum value spend by a single user in a single day. Then it will block the calls if this value is exceed by the specified multiplier)

Configuration:

- checkmaxdyndailycredit: 0=disabled,1=enabled for enduser,2=enabled for endusers and traffic senders
- maxdyndailycreditforenduser_min: will not block endusers if spend below this value (default is 10)
- maxdynmaxdailycreditforts_min: will not block traffic senders if spend below this value (default is 100)
- maxdyndailycredit_multiplier: max grow allowed within a single day. Default is 7 (you might lower it for better protection, however values below 4 is not recommended because that can be considered as normal calling pattern)

The disconnect reason for such calls can be configured by the "quotadiscreason" global config option. Default is 403.
Admins can receive emails about user blocks if the sendadminemailnotifications global config option is set to 2.

For example if user A normally spend 10 usd maximum per day, and if one day it suddenly reach over 70 usd then calls are blocked. Please note that smooth increase if traffic is still fine (the max allowed grows automatically if the user is below the max grow limit).
Another example:

- Monday (firs day) the user spend 10 usd. Fine, the maxdyndailycredit is set to 10.
- Tuesday the user doesn't make any call. No changes.
- Wednesday the user spends 5 USD. No changes since 5 is lower than the previous max which was set to 10 on Monday.
- Thursday the user spend 30 USD. Fine since 30 is lower the maxdyndailycredit * maxdyndailycredit_multiplier (which is 70). The maxdyndailycredit is set to 30.
- Friday the user (or a hacker which have stolen the user credentials) start to make mass calls or calls to expensive direction. Once his spending reaches 210, the account is blocked (210 = the previous maxdyndailycredit which is 30 multiplied with the maxdyndailycredit_multiplier which is set to 7)

Follow these steps to enable this for old database versions:

change tb_users.todaycredit data type from int to decimal(18, 5)
exec sql: ALTER TABLE tb_users ADD [maxdailycredit] [int] NULL
exec sql: ALTER TABLE tb_users ADD [maxdyndailycredit] [int] NULL
add a.todaycredit,a.maxdailycredit,a.maxdyndailycredit to [v_checkuser] select list
set globan config:
maxdailycreditforenduser=1
maxdailycreditforts=1
maxdailycreditperuser=1

4.13.20. Daily/monthly call duration limits

You can apply a maximum daily and/or monthly limit for call duration per user with the following settings:

- maxslperday (max allowed speech length per day in seconds)
- maxslpermonth (max allowed speech length per month in seconds)

When these fiels are set for a enduser or traffic sender then the initiated call durations are counted (preventing the caller to make more calls then these limits).
When these fiels are set for a SIP server then the terminated call durations are counted (no more calls will be routed to the SIP server above these limits).
Calls that was already started will not be interrupted when the limit is reached (the next call will be blocked).

You might set the calcmaxsl global config option to 1 to enable this feature (-1: means auto detect, 0: means disabled, 1: means enabled).

4.13.21. Fraud calls

By default the mizu server blocks most of the well-known satellite and premium numbers (blocksatellitecalls and blockpremiumnumbers).
You can also block other unwanted destinations from routing or billing

4.13.22. Blacklists

With the fix/dynamic and smart blacklist modules you can efficiently block unwanted numbers. See the Blacklisting/Access Lists chapters in this documentation and/or search for “black” in the Configurations form from MManage for more details.

4.13.23. Auto ban

Auto ban can be used to block unsolicited or telemarketing calls by rejecting the caller when bad statistics are detected.
See the Caller banning FAQ point for the details.

4.13.24. Billing and profitability

To make the billing strict (proof protect), you might also set the following global parameters:

- blocknotbilledcalls: 0=no (default),1=if best match packet is not found,2=if no exact packet match with prefix,3=block if not assigned directly for the user,4=check also tariff list prefix,5=check also parent tariff list
- blocknotprofitablecalls 0=no (default),1=yes,2=yes also if equal,3=block also when it is not set
- vgetpriceexactmatch: 0=no (default), 1=yes for resellers, 2= yes for all
- notbilledcallerr=0 0=default (error with 2 priority),1=error,2=critical
- blockhighcostcalls: don't route the call if the cost per minutes is higher than this value (default is 0 which means no limit)

Note: In the price lists you can set the * prefix with a very high cost (to be blocked by the above settings) or don't set at all to avoid wildcard matching.
(* prefix means any other prefixes which you haven't explicitly set)

You can implement call blocking to certain destinations by using "null routes".
Just create new routing patterns ("Routing" form) and don't set any destination for them (empty right side list).
This way the calls that matches your definitions will be blocked.

4.13.25. Alerting

Admins and support users can receive email, sms and/or push notification alerts about important events as described here.

4.13.26. Security checklist

The Mizu softswitch comes with default secure configuration, but extreme secure settings might prevent comfortable usage and we had to think about a good balance of security and usability. You might go through the below points to further secure your VoIP environment. Do not attempt to change all these settings, only those which are relevant for you.

Network security:

- make sure that your router or gateway performs proper ingress filtering (blocking of packets from outside the network with a source address inside the network).
- set proper firewall rules to protect your services (not important for the VoIP service itself, but you might host other services within your network which might require firewall protection)
- set proper packet flood filtering rules
- be aware of DDoS attacks and its mitigation especially if you are a big organization or VoIP is vital for you (while the VoIP service is capable to mitigate DDoS attack by itself, this will not help if your ethernet ports are fully flooded). Use a network provider capable to mitigate DDoS attacks.
- disable all unneeded network services such as NetBIOS and related services
- use secure access (HTTPS for websocket, TLS for SIP signaling, DTLS/SRTP for media, tunnel for admin access)

OS security:

- do not install any third party software on your VoIP server and don’t use it’s desktop for your daily work
- run MManage -> Config -> OS -> Secure Windows
- enable the embedded windows firewall. It’s speed and application level packet filtering is perfect for VoIP. Enable only the needed applications (mserver.exe, mssql.exe, vfpt.exe, tlsproxy.exe, mizuweb.exe)
- disable all unneeded network services (File and Printer Sharing, IIS, FTP, etc. With a clean windows install this is not necessary since not installed/enabled by default)
- disable NetBIOS over TCP/IP and Client for Microsoft Networks on network connections where it is not needed.
- don’t install any virus scanner (it is meaningless)

- choose a strong password for your OS accounts (especially for the Administrator account and better if you don't create or disable all unneeded OS user accounts)
- rename the Administrator account
- run the Microsoft Baseline Security Analyzer tool
- change the RDP port, preferably to 60089
  (registry: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp)
- use the Tunnel for remote access (built-in encrypted + compressed secure access)

DB security:

- choose a strong long password for your database access (especially for the sa account).
- change the default MSSQL port from 1433 (this is not a security issue, but when the MS SQL runs on the default ports, you can experience lots of login attempts).
- restrict the SQL service account (especially if your server is not dedicated to VoIP / you are using the server also for other purposes)
- create users with different roles and set only minimal rights as required (especially if the engine is hosting also other databases)

VoIP service security:

Check these parameters (don't set them blindly as some of these parameters might turn the server completely unusable for your use-case):

- strengthen the above listed configurations (DDoS attack prevention, per IP call limits, max monthly call and credit limits, daily credit limits, max line per user, etc)
- request (and verify) as much data from your subscribers as necessary (country, email address, etc)
- increase minpwdlength (default value is 4)
- set the fwdtootherdomains global config option to 0 to disable arbitrary call routing to other domains
- set the apiv2key global config option to a random value (a short 5 digit number will fulfill its reason)
- set the enforcestrongauth global config option to true (true by default since v.5.2 2013)
- set strongdigestauth to 1 (you might set it also to 2 but this is too much usually and not compatible with some devices)
- set the allowcalleridchange global config option to 0
- set the useshadigestauh global config to 2 to use SHA-256 instead of MD5 (if clients supports this)
- set the "allownoauthfork" global config option to 3 or 4
- set the "stricttransport" global config option to 2 or 3
- make sure that all users have a strong password (this should be fulfilled automatically if the rest of the settings are set correctly)
- set authfastusers to 1
- set remoteadmin to 0,1,2 or 3
- set max amount, max ports for users
- set maxmonthlycredit per user (as discussed in above sections)
- set callingcardauth to a more strict check (for example set to 9 for pin code strict only)

- configure the maxdyndailycredit global config option and dynamic credit/spend limits as discussed in above sections
- set the validateinput global config to 3 or more (0=no,1=basic without sql check,2=basic with sql, 3=normal,4=more,5=extreme)
- set the validatepassword to 2
- set the autocreatereguserstrict to 1
- set the securepasswords parameter to 1 or more
- set a random serverapikey (don't change it if you already have customized softphones)
- set publicservices to 0
- set the uauthverifypwd to 0
- set blockotherdomains to 1
- set fs_restrictsiptolocal to 3
- set blockscanning_treshold to a positive low value such as 10
- set the forwardclientaddress to 0 (to prevent forwarding the peer location)
- set the apisipauth to 1 (to enforce SIP authentication for API access when appropriate)
- set allownoauthcalls to 0
- set allowanonymouscaller to 0
- set securemove to 1 or 2 to prevent 301/302 call forward routing to traffic senders
- set reuseep to 0
- set reuseepauth to 0
- set reusedifferentportreg to 0
- set reuseupperregep to 0
- set cacheregistrations to 0 or 1
- set enablefirewall to 5
- set autoban to 2
- set earlystart to 0 and/or blockearlymedia to 1/2/3 and/or blockearlymedia_in to 1/2 if you wish to block all early media (including inband ringtones)
- leverage TLS security (optional)
- remove the "Lookup A number" option from Traffic senders if it is not a DID provider for users
- disable unwanted directions (unwanted countries, etc)
- check your traffic on a daily basis or setup scheduled tasks to check abnormal traffic amounts and notify you by sms or email. Use the Analyze form regularly.
- setup the allowed ip and trusted ip global config option properly by the apiv2ipauth global config option (applied especially for the http api). Disable trustlanip from global config if not needed.
- set trustsubnet, trustauthip and treatusersasknownip to false (these are false by default)
- set the allowtrustedip to 1 or 2
- set simultancallscreditcut to 3
- set the blocksatellitecalls and blockpremiumnumbers to true

- check proper dtmf input in your ivr scripts
- check your global configuration parameters related to IP, port, currency, credit amounts and other restrictions
- make sure that the A number authentication is set correctly if you are using IVR call forward. disable if not needed (turn off for each traffic sender at the "Functions" tab and/or set the anumberhandling global config option to 0)
- you should verify first payment manually (don’t accept automatically first PayPal payment from not verified users. see the PayPal settings)
- disable JSONP to other servers by setting the enablejsonp global config option to 1
- set handlerefer to -1 or -2 if you don’t need call transfer
- disable conference rooms if not needed (set enableconferencerooms to 0)
- you might set the autofinetune to 0 to set consistent service level and disable runtime feature set changes (but this is not optimal for servers with occasional high-load)
- search for billing related configurations: search for “credit”, “bill” and “pric” in the config wizard
- disable http file upload if no file sharing services are used by setting the allowhttpfileupload global config option to 0
- set apidefneedlogin to 3 for nonce verification on API access
- set apideftrustadmin to 4 (0: no, 1: admin if trusted ip (def), 2: always ip only, 3: always pwd only, 4: always ip+pwd)
- you can also apply system level limits by the licensecfg global config option (see the “License limitations” FAQ section)
- set the default credit to 0 by executing the following queries (from the direct query form or from SQL management studio):

  ```
  alter table tb_users drop constraint DF_tb_users_Credit
  alter table tb_users add constraint DF_tb_users_Credit default 0 for Credit
  ```
- if you have reached here, maybe you are interested also in the performance related settings (see the “Performance optimizations -fine-tune” FAQ)

4.14. High availability

With the Mizu VoIP you can implement a highly available VoIP service with automated backups and auto failover on all levels.
Please visit the following documents for the details:

4.14.1. Large scale VoIP

4.14.2. HA VoIP service

4.14.3. HA database

4.14.4. Database failover

4.15. Maintenance

When properly set up, Mizu software doesn’t need too many administration tasks. The routing will adjust automatically to the external conditions. Every software module has auto repair and cleanup features. Backup and scaling is automatically provided for you if you are using the Mizutech SaaS platform (Hosted VoIP solutions).

However if you have lots of traffic then you may need to watch the system parameters closely, setup proper alerting thresholds and adjust system setting as required.

4.15.1 Server configuration checklist

Software’s

Windows 2003/2008/2012/2014/2016/2019/2022/2025 server, MSSQL (Standard or Express) 2000, 2005, 2008, 2012, 2014, 2016, 2017, 2019, 2022, 2025

Firewall

-allow servers, FTP, IIS, MSSQL, SMTP, remote desktop (3386,3389)
-block: all others, file sharing, ms network

Services and memory

Auto restart critical services
For more than 3 Gb set boot.ini: /3GB /PAE on 32 bit systems

Licensing

Setup configuration with the proper software(s)

Config

Setup FTP and Voice directory NTFS access

MSSQL

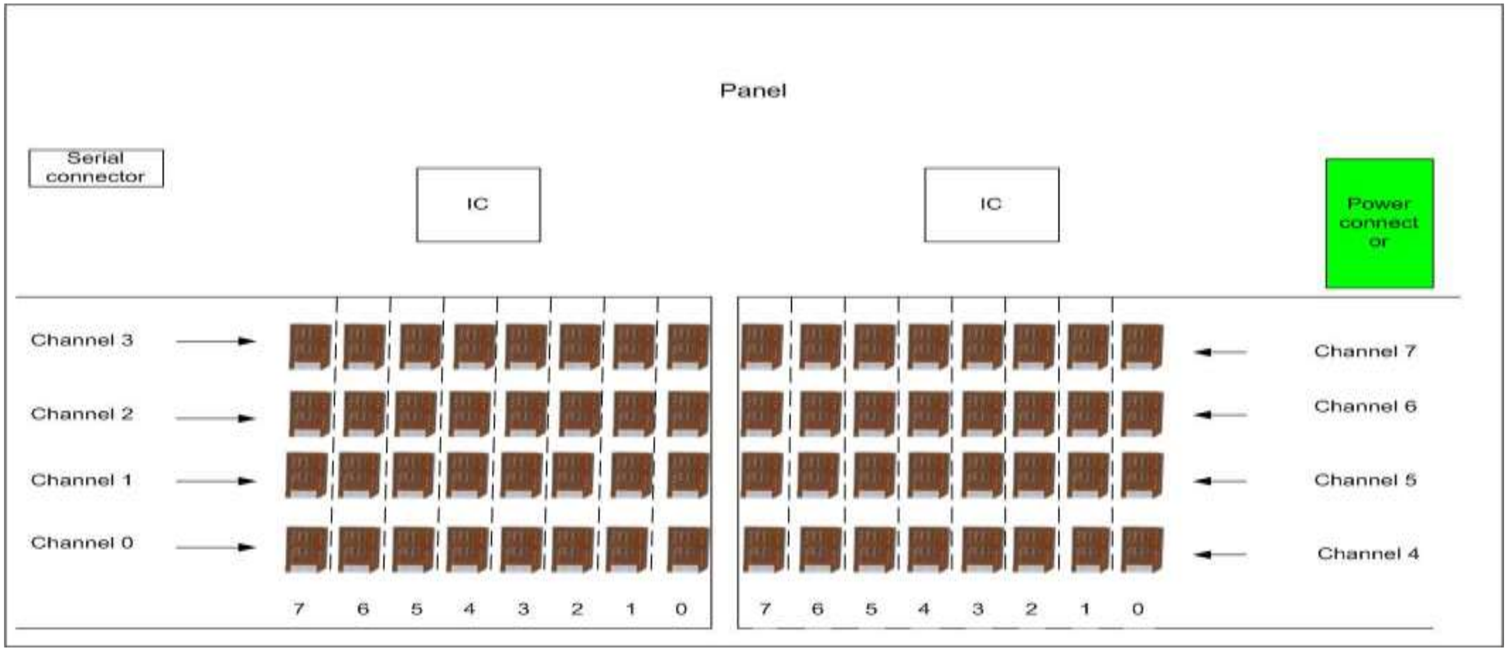
Change port to 2223
Enable pipes
Check configuration in Management Client

4.15.2 Gateway quick setup

Skip this chapter if you are not using VoIP-GSM gateways.

In order to get a working system, here is a checklist which may help you:

1. Connect the gateway(s) and/or the server to the network.
2. Install the MManage programs in a separate PC used for monitoring your Mizu devices. network (you can find it on the Mizu install CD which is shipped with every product)
3. Set up the gateway(s) and/or the server network parameters with the VnetCfg utility
4. Put your simcards into the gateway (see the image below)



5. Connect to the gateway or server with the MManage (by typing its ip address and username/password in the login form)
The default username/password is admin/tpwdadmin
6. Set up the basic parameters from the "Configurations" form
Be careful.
7. Set up one or more packets for the simcards in the "SIM Packets"
Be careful with the following settings: *prepaid/postpaid, allowedpartners*
8. Set up the simcards. You can add simcards manually, but it's easier to wait for them to register. Then you only have to modify its packets, owners and the recharging settings ("Simcards" form)
9. Add some traffic sender in the "Users and Devices" form.
Be careful with the authorization settings
10. Set up the routing ("Routing" form)
Add at least one routing pattern (name it as you wish)
Add at least one entry to its priority list (your newly created packet or some other direction)
11. Set up advanced routing –Optional
Firewall, prefix rules, BRS, etc
12. Set up the billing module –Optional
13. You are ready to accept traffic now.

4.15.3 Daily Maintenance

You should check at least the followings every day:
-Current Calls –to quickly check if you have the required amount of traffic
-Quality Statistics by traffic senders and terminating gateways
-Run a global system analysis (“Analyze” form)

4.15.4 Monthly Maintenance

-Check your cash flow (“billing” form) to check if your routing is still profitable
-Logs (errors and critical levels)
-Analyze your traffic by using the “Statistics” form
-Remove blacklisted but good numbers (if you are using blacklisting)

4.15.5 Server backup, recovery and maintenance

About
Most of the maintenance tasks are done automatically in background low priority threads and/or scheduled to offpeak times.
These followings are handled automatically:
-database tuning
-user maintenance
-background statistics
-table or record level backup to _backup database
-delete of the old records (for example old log entries)
-database backup to a predefined directory
-and many others

Since all data is stored in the SQL database, you have to protect only your database (the program directory can be recreated anytime by simple file copy or reinstall. Only the database path is stored here in the mizuserver.ini file which need to be reentered correctly after a restore).
Although there is no special maintenance needs with the mizu server, to have a working backup is a very important task to be set and tested properly. The mizu server will automatically create backups on the same server (without any additional settings), but for a real backup you have to change the backup directory to another physical location.
For more ratability you should use the MS SQL backup related capabilities instead of using the mizutech built-in backup methods.

What is a backup?

A backup refers to making copies of data so that these additional copies may be used to restore the original after a data loss event. Backups are useful primarily for two purposes. The first is to restore a state following a disaster (called disaster recovery). The second is to restore small numbers of files or records after they have been accidentally deleted or corrupted.
There are also many different ways in which the data storage devices can be arranged to provide geographic redundancy, data security, and portability.

Any backup strategy starts with a concept of a data repository. The backup data needs to be stored somehow and probably should be organized to a degree. Different repository models have different advantages. This is closely related to choosing a backup rotation scheme.
Unstructured: An unstructured repository may simply be a stack of floppy disks or CD-R/DVD-R media with minimal information about what was backed up and when. This is the easiest to implement, but probably the least likely to achieve a high level of recoverability.
Full + Incremental: A Full + Incremental repository aims to make storing several copies of the source data more feasible. At first, a full backup (of all files) is taken. After that, any number of incremental backups can be taken. There are many different types of incremental backups, but they all attempt to only backup a small amount of data relative to the full backup. Restoring a whole system to a certain point in time would require locating the full backup taken previous to that time and the incremental backups that cover the period of time between the full backup and the particular point in time to which the system is supposed to be restored. The scope of an incremental backup is typically defined as a range of time relative to other full or incremental backups. Different implementations of backup systems frequently use specialized or conflicting definitions of these terms.
Differential: A differential backup copies files that have been created or changed since the last normal or incremental backup. It does not mark files as having been backed up (in other words, the archive attribute is not cleared). If you are performing a combination of normal and differential backups, restoring files and folders requires that you have the last normal as well as the last differential backup.
Continuous data protection: Instead of scheduling periodic backups, the system immediately logs every change on the host system. This is generally done by saving byte or block-level differences rather than file-level differences. It differs from simple disk mirroring in that it enables a roll-back of the log and thus restoration of old image of data.

Built-in database backup

You can configure your database engine to do the backup tasks. If you don't have this possibility, the Mizu Server can also make database backups. Set the following configurations:
dbmaint_backuplevel: 0=no backups,1=daily,2=daily,monthly,weekly,3=hourly,daily,monthly,weekly,4=keep lots of files
dbmaint_backupdbdir: local path (accessible for the server)
dbmaint_backupdbnetworkdir: path accessible for the database engine (it is necessary when the database is located on a separate box. Otherwise can be left empty –default to dbmaint_backupdbdir)

Using backup database tables

To keep your active size smaller, you can move the content of big database tables to another database. For example cdr records older than 5 month, etc. This backup database will be easily accessed by the MManage.
dbmaint_backuptables: backup cdr records and other tables to xxx_backup: 0=no,1=cdrs,2=extended,3=all

Frequently used and important tables are backed up daily, the other monthly.
The amount of data that is backed up can be controlled by the "**dbmaint_removecdrs**" (measured in days. is direct proportional with this setting)
If the dbmaint_backuptables is set to 2, than the amount of data can be controlled by the *dbmaint_removeother* value.
Logs are removed after *dbmaint_removelogs* days.

The backup databases are created with the same name as the current database with a "backup_" prefix.
If you want to place the backup tables to a separate server, you must set that backup server as a linked server in the main database and set the "**linkedserver**" config option in the server inifile.

The backup intervals can be controlled by the "**dbmaint_backuplevel**" option: //0=no backups,1=daily,2=daily,monthly,weekly,3=hourly,daily,monthly,weekly,4=full,5=keep lots of files
(be aware that if you set it to 0, than the other option will have no effect)

In order to take advantage of the backup database, you must set the "Secondary database" option in **MManage** to "Autodetect" or "Secondary DB". When set to autodetect, it will attempt to load data from the backup database if the query date is set to more than 1 month later (it will failback to main database if is set to earlier than 2 week). In this case the "until" date is modified automatically to yesterday midnight when needed.
For the MManage to be able to connect to the backup database, the "**backupdatabaseaddress**" must be set up correctly when a separate database is used.

Database maintenance

The server will optimize the database engine automatically.
dbmaint_do: enable/disable the daily and the monthly maintenance
dbmaint_removelogs: Trunk log tables and remove logfiles after x days

Saving recorded voice

Serverftpvoice: where to store recorded audio
Serverftpdailydir: set to true to create separate directory for recorded voices daily
Keeprecorded: days to keep voice records
Voicebackupdir: backup your recorded voices to this secondary location
Keepbackuprecorded: days to keep the voice backups

The voice recording option can be set for any user by checking the "Record" checkbox on the user configuration form in MManage.
Conversation will be saved in the directory specified by the "serverftpvoice" global config option.
A separate backup can be created in the directory specified by the "voicebackupdir" global config option.
Out of date recorded files can be deleted by setting the "keeprecorded" option accordingly (days to keep).
Recorded files can are compressed and encrypted by default.
On new server installation, make sure that the voice directory are accessible via ftp for the MManage (for listening on the "CDR Record" form). To make things easier it is preferable to setup the ftp passwords the same as the database login.

Alternative backup deletion

You can define an alternative backup deletion method by the following values:
Deldbbackup: delete old backup files after this day elapsed
Dbbackupdir: delete from this directory and its subdirectories

You can automate backup cleanup by setting the following global config values:
Deldbbackup: days to keep (-1 disables cleenups)
Dbbackupdir: database backup directory
dbdelbackupdir1, dbdelbackupdir1, dbdelbackupdir3: database backup subdirectories
This feature is useful, when the database engine don't have cleanup feature.

Disaster recovery

You must always have a working recovery plan.
Here is a template with dual server configuration:

if the application server fails (the server directly connected to the internet, with your public ip

1. call your ISP support to change the internet cable to the backup server, and when it will be available connect to the "backupserver" with the remote desktop "root" account
-on the backup server do the following:
2.   enable the "mserver" service
3.   launch the start batch file (from gk directory)
4.   check the vservdebuglog and the MManage

if the backup server fails (the server behind the main server, with private ip)

-connect to the main server with the remote desktop "root" account
-On the main server, do the followings:
1. launch the stop batch file (from the gk directory)
2. Enable and Start the SQLSERVER service
3. Restore latest database
4. launch the start batch file
5. check the vservdebuglog and MManage (you must have current calls)
6. you are ready

MSSQL Server and MSSQL Studio

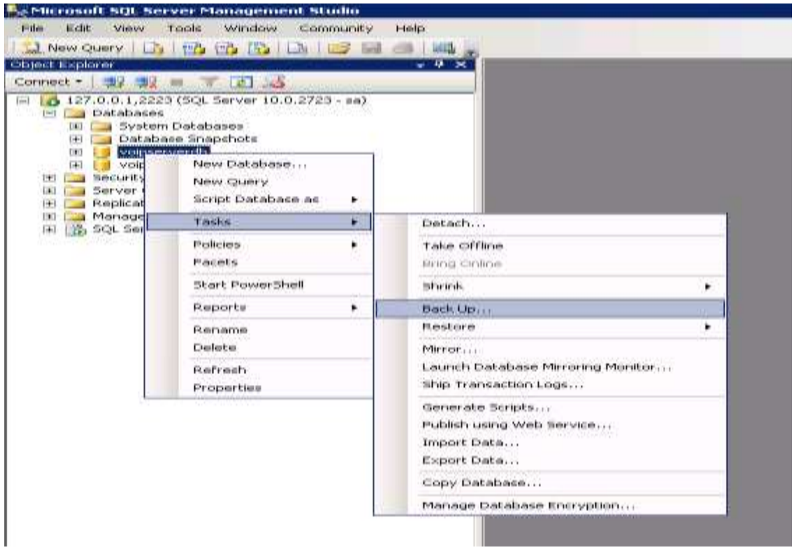
Microsoft SQL Server is a relational model database server produced by Microsoft. Its primary query languages are T-SQL and ANSI SQL. You can work directly with the database using the client IDE tools, and several complementary systems that are packaged with SQL Server. These include: an ETL tool (SQL Server Integration Services or SSIS), a Reporting Server, an OLAP and data mining server (Analysis Services), and several messaging technologies, specifically Service Broker and Notification Services.
SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server.

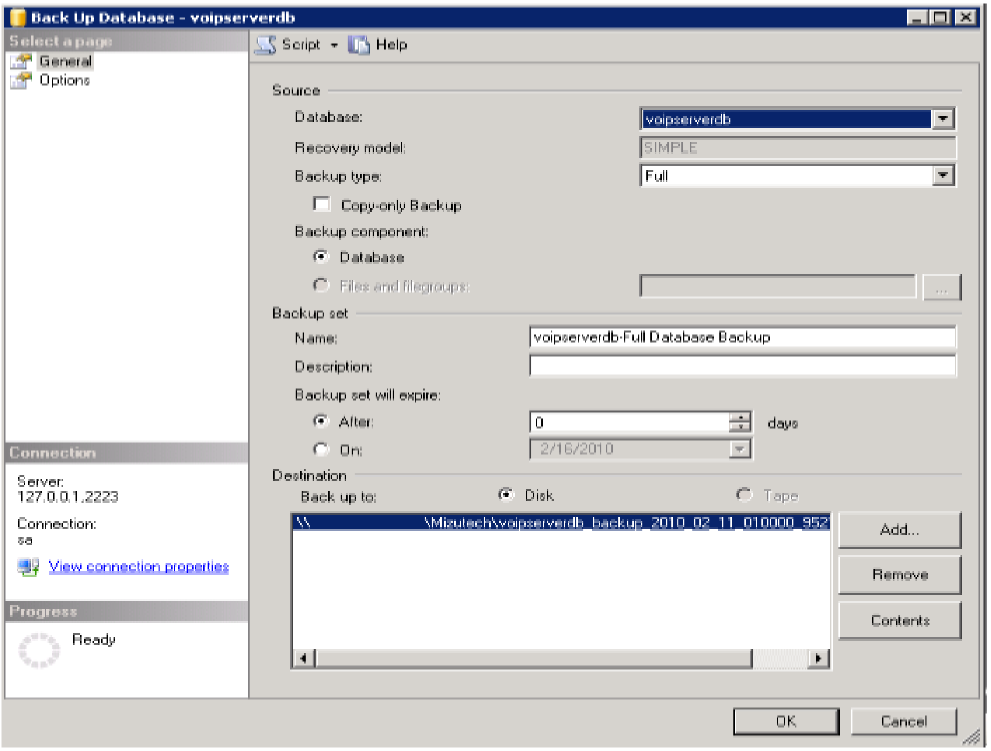
A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.
SQL Server Management Studio can be downloaded for free from the Microsoft Website:
http://www.microsoft.com/downloads/details.aspx?FamilyID=08E52AC2-1D62-45F6-9A4A-4B76A8564A2B&displaylang=en

How to make a manual backup

First open Microsoft SQL Server Management Studio, open Object Explorer, open Databases, right-click with your mouse on the database you want to back up, select tasks and select Back Up from the menu.

The next window will appear:





Source
Database: voipserverdb
Recovery model: SIMPLE
Backup type: Full
☐ Copy-only Backup
Backup component:
● Database
○ Files and filegroups:
Backup set
Name: voipserverdb-Full Database Backup
Description:
Backup set will expire:
● After: 0 days
○ On: 2/16/2010
Destination
Back up to: ● Disk ○ Tape
\\ \Mizutech\voipserverdb_backup_2010_02_11_010000_952

Server:
127.0.0.1,2223
Connection:
sa

Ready

At Source/ Database select the database that you want to back up.
At Backup type select either full for full type of backup or differential, for differential backup.
At Destination click on the Add button, select the disk, and path to the file where you want to save the backup file. If this is on another

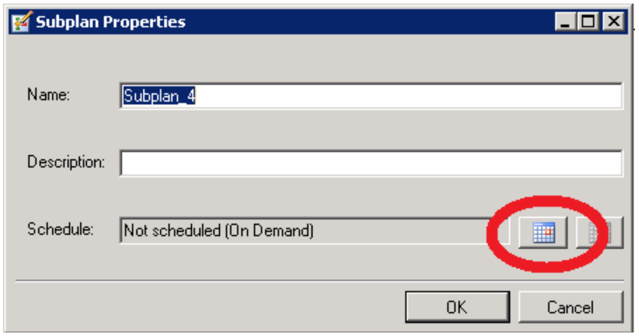
server type \\ double-slash, followed by the IP address and path on the another machine.

How to set up automated backup

In the Object Explorer open Management, under Management right-click with your mouse on Maintenance Plans and select New Maintenance Plan. In the small window that appears give it a name.

Click Add Subplan, Introduce names, description, and set the schedule for the given Subplan, by clicking on the small calendar icon.

The Schedule settings window looks like this:

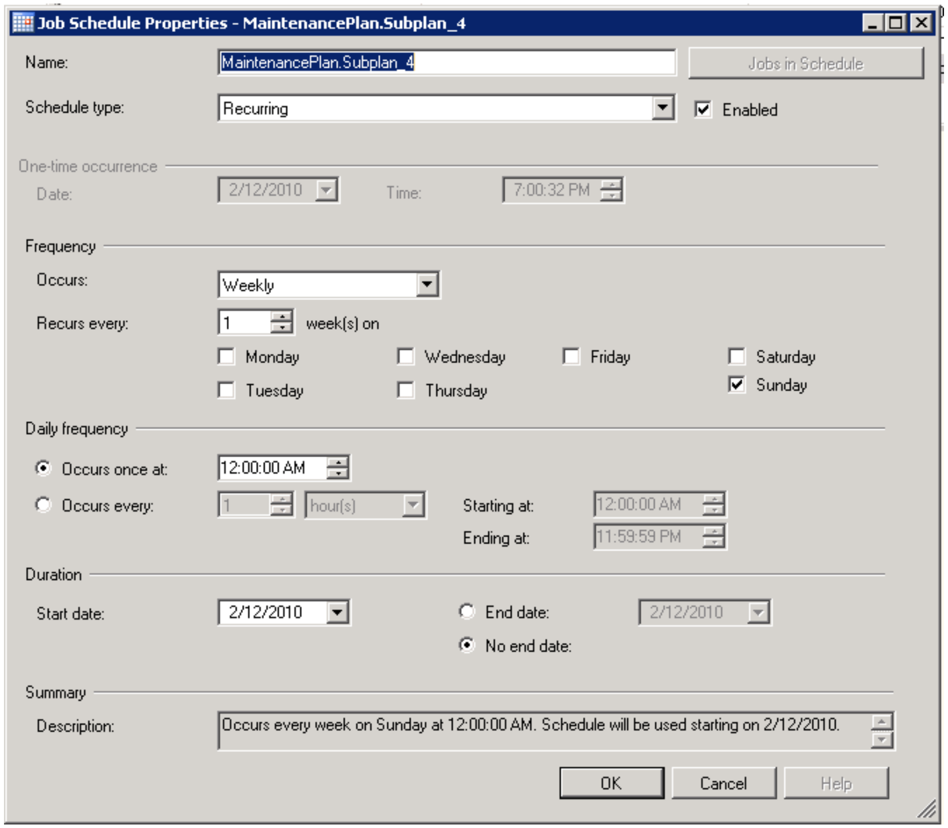


Subplan Properties

Name: Subplan_4

Description:

Schedule: Not scheduled (On Demand)



Job Schedule Properties - MaintenancePlan.Subplan_4

Name: MaintenancePlan.Subplan_4

Schedule type: Recurring ☑ Enabled

One-time occurrence

Date: 2/12/2010 Time: 7:00:32 PM

Frequency

Occurs: Weekly

Recurs every: 1 week(s) on

☐ Monday ☐ Wednesday ☐ Friday ☐ Saturday

☐ Tuesday ☐ Thursday ☑ Sunday

Daily frequency

● Occurs once at: 12:00:00 AM

○ Occurs every: 1 hour(s) Starting at: 12:00:00 AM

Ending at: 11:59:59 PM

Duration

Start date: 2/12/2010 ○ End date: 2/12/2010

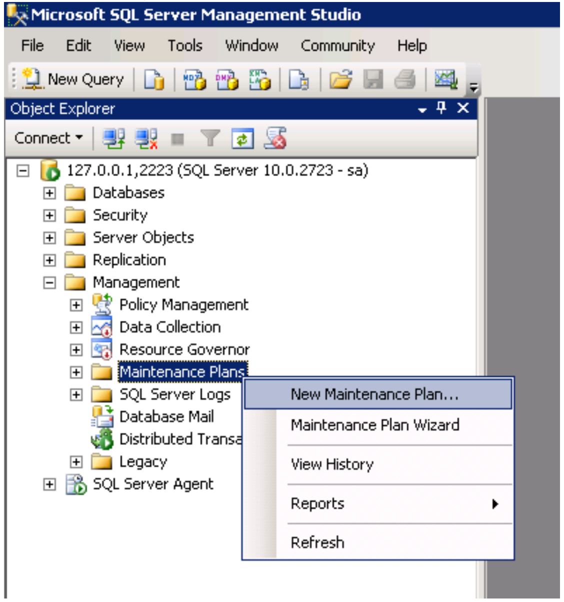
● No end date:

Summary

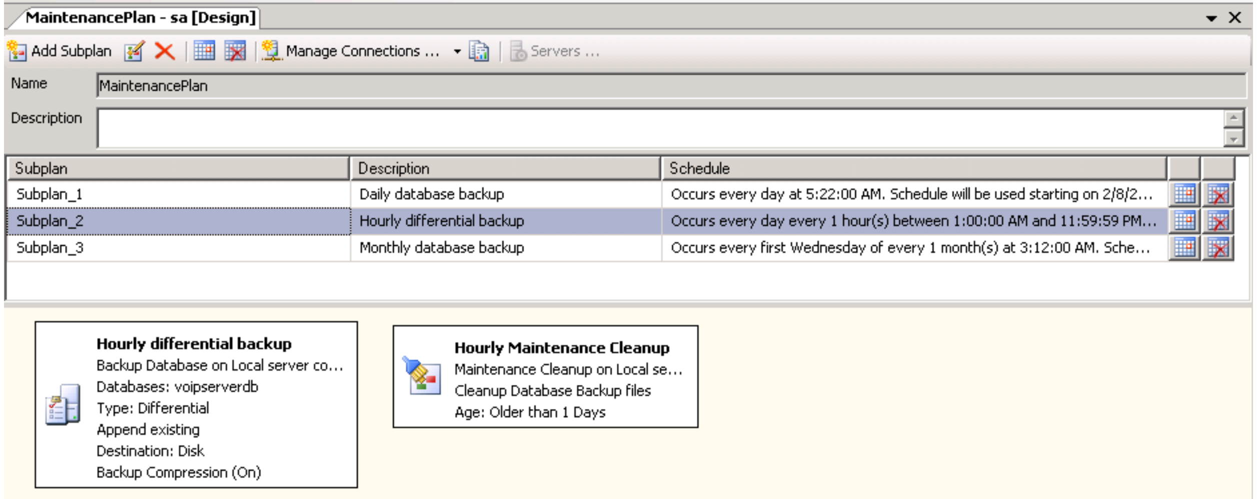
Description: Occurs every week on Sunday at 12:00:00 AM. Schedule will be used starting on 2/12/2010.

Once the time and frequency of a given schedule is set the tasks from the left side of the window can be dragged and dropped in the field under the subplans list. For every subplan a ‘Backup database task’ and a ‘Maintenance cleanup task’ can be dragged/dropped and defined on this field.

For example you can create 3 folders in the shared folder on the application server, one for hourly differential backup which is backed up once every hour, and the backup is deleted once every day. You can create another for a full daily backup, and one for full monthly backup.



Once you drag/drop and define the backup tasks from the lower left pane to the right, the subplans will look like this:



Name: MaintenancePlan

Description:

Subplan	Description	Schedule
Subplan_1	Daily database backup	Occurs every day at 5:22:00 AM. Schedule will be used starting on 2/8/2...
Subplan_2	Hourly differential backup	Occurs every day every 1 hour(s) between 1:00:00 AM and 11:59:59 PM...
Subplan_3	Monthly database backup	Occurs every first Wednesday of every 1 month(s) at 3:12:00 AM. Sche...

Hourly differential backup
Backup Database on Local server co...
Databases: voipserverdb
Type: Differential
Append existing
Destination: Disk
Backup Compression (On)

Hourly Maintenance Cleanup
Maintenance Cleanup on Local se...
Cleanup Database Backup files
Age: Older than 1 Days

http://support.microsoft.com/kb/930615

Dual Server Configuration

In the Dual Server Configuration there are two separate servers used, the application server and the database server. The MizuTech VoIP Service is running on the application server and usually a Microsoft SQL Server database is installed on the database server. The direction of the backup is from the database server to the application server, into a shared folder, so that if the database server experiences problems the other server can provide the latest backup.

In an ideal case, backup is of three kinds: an hourly differential backup and a daily full backup about the voipserver database, and a monthly full backup about all databases. For more ratability you can also setup continuous backup (with replication or log sipping)

Tools and scripts

Scheduled backup tasks for MS SQL Express: http://www.diaryofaninja.com/blog/2011/02/14/howto-quick-amp-dirty-sql-express-scheduled-backup

Fee backup & FTP tools (for one single database): http://sqlbackupandftp.com/download/
http://www.sqlbackupmaster.com/download
Script for SQL backup & FTP: http://www.morrenth.com/script-to-backup-zip-and-ftp-sql-server-database.aspx

5. FAQ

Frequently Asked Questions

5.1. Abbreviations

Common terms used in this document:

Caller: The person making/initiating a call
Callee: The person receiving a call (called party)
Global config: server wide configurations which can be edited from MManage “Configurations” form and stored in the tb_settings table
ASR: average success ratio (percent of the connected calls)
ACD: average call duration. The same as ACL *(ACD can also mean Automatic Call Distributor, but this is not used here)*
ACL: average call length. The same as ACD
SIMID: sim identifier. 13-17 digit number stored in the simcard (and written on the simcard)
IMEI: gsm engine identifier (should be globally unique)
ACT: average connect time. The time elapsed from setup until the connect in seconds
PF: profit. (for correct values, requires your billing module to be properly configured)
SUCC: successful call count (same as ASR but not in percent)
CCC: concurrent (simultaneous) call count
CC: callcenter
TCP: is a connection-oriented internet protocol
TLS: encrypted TCP
UDP: internet core protocol for datagram packets (not reliable)
RTP: media channel protocol
SIP: The Session Initiation Protocol (SIP) is a signaling protocol used for establishing sessions in an IP network. A session could be a simple two-way telephone call or it could be a collaborative multi-media conference session.
WebRTC: Web Real-Time Communication used implemented in HTML5 browsers and other clients
H323: H.323 is an ITU (International Telecommunications Union) recommended standard, which provides a foundation for audio, video and data communications on non-guaranteed Quality of Service networks
RAS: used in H323. Used between the endpoint and its Gatekeeper in order to
Allow the Gatekeeper to manage the endpoint (Registration, Admission, and Status)
GK Registration: Endpoint will send an RRQ and expect to receive either an RCF or RRJ
H225: Call Signaling is used to establish calls between two H.323 entities
H245: generally transmitted on a separate TCP connections by most older endpoints
REGISTRAR: server-side component that allows SIP REGISTER requests
IEC: international escape code

NEC: national escape code
AC: area code
NUM: phone number
ANI / CLI – Automatic Number Identification or Caller Line Identification
IVR – Interactive Voice Recognition
LCR: least cost routing (price)
BRS: best route selection (price + quality + other settings)
SRTP: media encryption protocol
ANI/CLI authentication: Automatic Number Identification/Calling Line Identification
Toll free: is a special telephone number, in that the called party is charged the cost of the calls by the telephone carrier, instead of the calling party. This can be configured as a normal access number (enduser) and eventually with higher billing (because we will be the billed party in this case)
local DID number: normal access numbers. Usually you will have separate DID numbers for different regions to minimize enduser costs
callback: DID or toll free number configured as enduser with iscallback set to the required IVR
ANI callback: same as callback with User-ID based authorisation (A number)
Virtual Numbers (DID): "real" phone numbers allocated for users. You have to buy DID numbers from CLEC or any other service provider like didx.net
SMS callback: callback triggered by received SMS message. You have to subscriebe to a two way sms service like clickatel.com. users can be authenticated by sender ID, pincode or username/password insterted to the sms text.

5.2. Text to Speech

TTS capabilities are available for the Mizu VoIP Server IVR system to play any text, without the need to have a recorded voice file for it.
You can use Microsoft SAPI or eSpeak (espeak.msi), configurable with the **tts** global config option, then use TTS commands in your IVR scripts after your needs. See the IVR guide for more details.

5.3. Spam calls

If your server is exposed on the public internet, then you can expect all kind of attacks, password guessing scripts and junk call attempts.
This is normal for public facing servers and should be mitigated automatically by the server. Just make sure that you are using strong passwords, including for the SIP accounts.
In case if your customers are connecting only by app/service, then you might use non-standard ports (for example port 5070 instead of the default 5060) and/or you can restrict the apps after their User-Agent with the alloweua global config.
See the security guide for more details.

5.4. How can I make test calls?

1. simply right click on a channel (“Simcards” form) and select the “Test call” option

2. or use one of the voip clients from the “Config” menu -> Utilities

5.5. How to check the call quality on a specific channel?

1. In the “Set Directions” box set the preferred simid. Then go to the “Statistics” form and check the ASR/ACD values.
2. Start some test calls (right click on the preferred channel and then hit the “Test Call” menu)
3. Listen to conversation. (“Voice Here” form)

5.6. Manually remove upper push registrations

This might be used if you have a gateway with multiple servers and you wish to wipe out push users to a certain upper server:

```
delete from tb_users where type = 0 and (contactname = 'DOMAIN' or contactname = 'IP' or hwid = 'DOMAIN' or hwid= 'IP')
```

5.7. Server Recovery (in a separate app and db server configuration)

<u>if the application server fails</u> (the server directly connected to the internet, with your public ip

1. call your ISP support to change the internet cable to the backup server, and when it will be available connect to the "backupserver" with the remote desktop "root" account

-on the backup server do the following:

2. enable the "mserver" service
3. launch the start batch file (from gk directory)
4. check the vservdebuglog and the MManage

<u>if the backup server fails</u> (the server behind the main server, with private ip)

-connect to the main server with the remote desktop "root" account

-On the main server, do the followings:

1. launch the stop batch file (from the gk directory)
2. Enable and Start the SQLSERVER service
3. Restore latest database
4. launch the start batch file
5. check the vservdebuglog and MManage (you must have current calls)
6. you are ready

5.8. No incoming calls (no new calls in current call list in peak time)

1. Check the logs (filtered to „Server”)
2. If you cannot find the solution then.
   a) Restart the server.

b) Call the administrator.

5.9. Calls in „routing” status

1. If all calls are in routing status, then restart the gateway.
2. If this behavior is specific only for some of the gateways, then check if you have enabled the voipgsmgw.exe and the vclientsrv.exe on the windows firewall.
3. If enabling this programs on the firewall and restarting the service (stop.bat, start.bat) will not help, then do a software upgrade and restart the PC.
4. If still in routing mode, then call the administrator.

5.10. SIP caller cannot call

1. Check disconnect reasons in cdr record for that caller
2. Check username/password
3. Check credit (if prepaid user)
4. Check caller techrefix, and the routing settings for that techprefix

5.11. SIP called cannot be called

1. Check disconnect reasons in cdr record for that called
3. Check if username exists
4. Check if usergroup matches the caller usergroup
5. Check user firewall settings

5.12. No voice (caller and called cannot hear each-other)

1. Check routertp settings for the caller and the called
2. Check called firewall and nat settings

5.13. Wrong disconnect reasons

1. Check firewalls
2. Check the log file for that directions

5.14. MManage cannot connect to the server

1. Ping the server box. If ping is working, then check your username/password
2. Restart the server if you are sure that it is blocked
3. If still is not working, call the administrator immediately

5.15. Too slow MManage

1. Check your internet connection
2. Check server processor load. If too high, then check server logs, and if necessary, restart the server
3. If the problem persists, call the administrator

5.16. Query user register state

Using the API, you can use the reguser request to get the register state of a user like this:
http://SERVERADDRESS/mvapireq/?apientry=reguser&usr=USER_TO_QUERY&authkey=APIKEY&authid=ADMINUSERNAME&password=ADMINPASSWORD&now=555

You can also request the register state of a user from the database using the following SQL query:
select top 1 id from tb_users where username = 'USER_TO_QUERY' and status > 0 and statusdate > getdate() - 0.1 and Enabled <> 0 and temporarydisabled <> 1
This will return the user id if the user is registered, or null/no record if the user is not registered, disabled or doesn't exists (status 0 means unregistered, 1 registered, 2 and 3 means user is in call).
Note: direct database connections are not available with the free/compact version.

Call decisions based on user register state usually can be handled by the server itself.
For example you can create a routing rule where you will add multiple target users (or a group of users) and the server can automatically route the call to the "best" available user (registered, not in call, longest time not routed and/or priority orders)

5.17. How to set advanced user settings?

There are many advanced user settings which can be applied for endusers/traffic senders/sip servers/other users and don't have a GUI element on the "Users and Devices" form. These are more rarely used settings with no exposed user interface controls to prevent cluttering the configuration pages, however you can still easily change them using one of the following ways:

- Users and Devices form -> Settings page -> Details link -> Edit field
- Drag the splitter right to the Users and devices tree view control until a grid appears (or click on the Details link -> Toggle grid link). Click on the "Search fields" link and enter the field name or a part of it then hit the Enter button. The grid will be reloaded displaying your field as the first column where you can edit it as you wish.
- Use a simple SQL query from the Direct Query form. For example: update tb_users set xxx = y where username = 'zzz'

5.18. How to set a random caller-ID?

The Caller-ID can be rewritten to a random number or to a random number with a specified prefix or to a random number from a specified list or by any other SQL driven logic.
You can use the Rules for this ("Rules" form).

Create a new rule with the following parameters:

- Run at: At routing end
- Conditions: after your needs
- Get Data: callernumber
- Action: Full replace
- Data to set: CallerNumber extended

The Action Value should be set to something like one of the followings:

+9222222222
This will generate rewrite the caller-id to the specified +9222222222 number (not random, mentioned here only for clarification).

+9222222222{SELECT FLOOR(RAND()*10)}
This will generate numbers between +92222222220 and +92222222229.

234{SELECT FLOOR(RAND()*90000000)+10000000}
This will generate numbers beginning with 234 and total length of 11 digits. Example: 23480635297.

Example to rewrite the called number to first 8 digit from the called number and 5 random digits using two queries in the Value string (because it is possible to use multiple queries):
{select substring('[callednumber]',1,8)}{SELECT FLOOR(RAND()*90000)+10000}

Example to rewrite the called number to first 8 digit from the called number and 5 random digits using one query:
{select substring('[callednumber]',1,8) + CAST((SELECT FLOOR(RAND()*90000)+10000) AS VARCHAR(32))}

To set a caller id from Phone Numbers, you can use this string for the rule:
{SELECT TOP 1 telnumber FROM tb_telnumbers WITH(NOLOCK) WHERE ttype=X and free=X ORDER BY NEWID()}

You might also create a separate table to store your numbers and load from there (instead from the tb_telnumbers table).
For example create a table named tb_calleridnumbers with one number field as varchar and use a query like this:
{SELECT TOP 1 number FROM tb_calleridnumbers WITH(NOLOCK) ORDER BY NEWID()}

It is also possible to specify endpoint keywors in square brackets. For example you might select a caller id which match the called number country like this:
{SELECT TOP 1 number FROM tb_calleridnumbers WITH(NOLOCK) WHERE '[called_norm]' like SUBSTRING(number,1,2)+'%' ORDER BY NEWID()}

In this query the [called_norm] will be replaced with the normalized called number and a random number will be selected with a match for the first two character (the country prefix code)

Another example in case if your tb_calleridnumbers has a prefix column which should match with the prefix of the called number (3 digits), ignoring the first digit (with other words: the second, third and forth character from the called number must match the 3 digit prefix column in your tb_calleridnumbers table):
{SELECT TOP 1 number FROM tb_calleridnumbers WITH(NOLOCK) WHERE prefix = (select substring('[callednumber]',2,3)) ORDER BY NEWID()}

5.19. Import/export data

Any data/rows can be imported/exported to/from the VoIP server database tables using one of the following methods:

- MManage -> File menu -> Export/Imports
- SQL Management Studio -> right click to database to bring up the popup menu -> Tasks -> Import…
- Using direct SQL (INSERT commands or BULK IMPORT) from MManage -> Direct Query form or from SQL Management Studio
- You can also import/export to a new temporary table first and from there insert to the final table.

5.20. User settings

Many configurations can be set on two levels: globally or per user.
For global configurations just use the “Configuration Wizard” (for the important settings) and “Configurations” form (for all settings).

Per user settings can be configured in the following way:
The most important settings can be set from the MizuManage user interface from the “Users and devices” form.
Other (more advanced / all) per user configurations can be set multiple ways:

- Users and devices form -> Expand the grid (scratch out or double click) -> Search field link -> Enter the field name or a part of it
- Users and devices form -> Details link -> Edit fields
- Direct Query form -> use SQL like: UPDATE tb_users SET fieldname = 'fieldvalue' WHERE id = 1234 (or WHERE username = 'user')

5.21. Create custom reports

You will need some minimal SQL knowledge to be able to create custom reports.
An easy way to create reports is to create something similar from MManage first, then go to Direct Query form and click on the "Last" link which will show the last SQL executed by MManage and then you modify it after your needs.

For example a call details report can be easily listed by just using the CDR form:

1. Open the CDR form
2. Select "All" from the drop-down list (instead of the default "Last 35")
3. Se the correct date - time interval (left-top time dropdown -> select "Custom" and then select start/end date)
4. Fields menu -> Filter -> Set Direction Filter: select Source enduser or traffic sender
5. Execute the query and then go to the Direct Query form, load it with the "Last" link and modify it after your needs

This is how such a query looks like (copy-paste it to the Direct Query form and execute):

```
SELECT b.datum as 'Date', b.realduration as 'Seconds', cast(b.costenduser as decimal(10,2)) as 'Cost', u1.username as 'Caller', b.callednumber as 'Called',
(select top 1  tb_directions.name  from tb_directions with(nolock) where b.dirid = tb_directions.id) as 'direction name'
from  tb_cdrs b WITH(NOLOCK)
left join tb_users u1 with(nolock)  on (b.callerid = u1.id )
WHERE
b.datum > '5/1/2025 0:0:0' AND b.datum < '5/31/2025 23:59:59' AND
b.callerid = X AND b.realduration > 0
ORDER BY b.datum
```

5.22. SQL to reset upperserver register routing

reset upper servers:

```
update tb_users set sendcfgcount = 0, lastrouted = getdate() - 1, hwid = '', ContractComment = '' where username in ('xxxx')
```

reset selfrest regival:

```
update tb_users set RegTimeout = 1 where RegTimeout > 1 and username like 'self%'
```

5.23. SQL to reset user multi-device addresses

Execute this query if the server is sending the calls (also) to old/wrong addresses for multihome users / call fork:

```
update tb_users set addrlist = '', transip = '', transport = null
```

5.24. Media Routing

The Mizu VoIP server includes advanced RTP routing capabilities including RTP/RTCP routing or offloading, voice/video/fax/data routing, NAT handling, DTMF processing, media transformations, RTP mixing for conference calls, transcoding, voice recording, statistics generation and more.

It is important to be aware that if all media is routed via the server this can be responsible for ~80% of the total CPU and ~95% of the bandwidth usage (so the signaling and all its processing requires only ~20% CPU and ~5% of the bandwidth), thus it is important to take attention on this and try to offload as many media routing as possible.

SIP endpoints are usually capable to rote the media between directly between them (peer to peer) and only the signaling have to go through via the server. Whether the endpoints is capable for direct RTP routing, depends on the followings:

- Endpoints are behind NAT or has a public IP (if one endpoint is public and the endpoints are not very dumb, then the RTP can be routed directly; if both endpoints are behind the same NAT then direct routing should always succeed if the endpoint can use private IP or the router can route back packets to its public IP)
- NAT type: how the router(s) or the NAT device(s) handles the NAT mapping as described here
- Endpoint capabilities: ICE/STUN/TURN usage, intelligent NAT handling, stable RTP ports, keep-alive, rport support, UPnP
- Firewall/access control rules: some gateways and other devices such as carrier equipment might be configured to accept the media only from the signaling address especially with IP based authentication

Common scenarios:

- When call is made between endusers, it is very possible that both of them are behind NAT and in this case the server should route the RTP except if the users are using advanced SIP clients or they are behind the same NAT.
- For inbound and outbound calls the trunk/gateway/service provider most probably is on a public IP thus the media should be able to pass directly.
- If some server side media processing must be performed (such as conference mixing, voice recording or DTLS decoding) then the media will be always routed, regardless of the settings.

The Mizu VoIP server can decide if RTP routing is necessary based on the server settings and the above conditions as detected by the server.
The media routing can be configured from MManage:

- globally from Config Menu -> Configurations -> RTP relay (will change settings for all users and default values for new user records)
- by user: from Users and Devices form -> Functions page -> Media relay -> Route RTP caller/called settings (mapped to tb_users.RouteRTPCaller and RouteRTPCalled fields)

The configurations are loaded and merged from the caller and the called party user. If the routertp setting is set to 1 or 7 and it will be applied regardless of the other endpoint setting, otherwise the more strict setting will be applied.

The following values are defined:

- 0: check caller/called settings: will load the config from the peer user settings. You should use this only for one of the settings such as for the RouteRtpCaller.
- 1: don't touch the sdp and the rtp: the SDP will be forwarded as-is. Use this if for some reason you don’t want the server to touch the SDP and the media or the server is doing something incorrectly. [Same as “Bypass” config]

- 2: sdp correction if necessary: the server might modify the SDP in a way to make the RTP direct routing possible such as changing private IP to public, inserting more candidates, fixing codec misconfigurations and handling known NAT related issues including known software dependent bugs. Use this only if you always wish to avoid RTP routing
- 3: route rtp if both behind nat: if both endpoints are behind NAT then RTP routing is usually necessary, especially if the endpoint doesn’t have advanced NAT handling capabilities such as STUN usage. This should be the preferred setting to offload as many media routing as possible. [Same as “Auto –Performance” config]
- 4: route rtp if caller is behind nat: rarely used, similar to 6
- 5: route rtp if called is behind nat: rarely used, similar to 6
- 6: route rtp if any endpoint is behind nat: a very conservative setting to make sure that RTP can always pass. Use this instead 3 if users are using dumb endpoints. [same as “Auto –Strict” config]
- 7: always route rtp: will route the media unconditionally. Use it if media routing is a must such as peer accepts RTP only from signaling IP or for topology hiding [Same as “Route” and “Force route” config]

The server can take intelligent decisions whether to route or offload the media only if the above setting is configured to 3,4,5 or 6. For example it will bypass the media if both device are behind the same NAT and direct routing should be possible.

Media routing should be avoided whenever possible.

The softswitch is capable to "intelligently" find out if it needs to route the media or not (to bypass the RTP routing whenever possible and route the RTP only when necessary).
You can reconfigure the default settings globally (for all users) from Config menu -> Configurations -> RTP Relay or by users from the Users and Devices form -> Functions tab -> Media relay.
The most conservative setting is the "Always route rtp", when MPUSH will always relay the media streams (the RTP should succeed in this case even if both parties are "dumb").
If you wish to avoid RTP routing by the server (to reduce CPU usage and shorten the RTP path), then you might set this to one of the followings:

- *“Route RTP if one peer is behind NAT”: this is fine if the other peer has good RTP NAT handling capabilities. For example if your server has advanced NAT handling, then you might set this option to it (SIP servers -such as Asterisk- has only very basic NAT handling capabilities)*
- *“Route RTP if both behind NAT”: RTP routing between endpoints are usually required, even if both peers has advanced NAT handling capabilities (because sometime they might not be able to determine their correct public address)*
- *“SDP correction if necessary”: Probably you should never use this (unless you know that the peers are always on public IP or both peers has advanced NAT handling)*

The recommended setting is:

- 3: for best performance and quality (99% of the calls should behave correctly and only dumb SIP endpoints might fail for which you might set to 7)
- 6: if you must ensure that no calls can fail due to poor endpoint side NAT handling
- 7: for restrictive trunks or topology hiding only

The advantages of media routing are the followings:

- NAT handling can’t fail

- Server side media handling capabilities (call recording, transcoding, etc. But if server side processing is needed, the media is always forced to be routed anyway.)
- Topology hiding (when you don't wish to reveal the other endpoint address)

The disadvantages of media routing are the followings:

- Higher CPU usage
- Higher bandwidth usage
- Longer media path which might result in more delay (if the server is not close to the endpoints or on the way between them)

5.25. Export CDR's

You can easily export CDR records by listing them on the CDR form (with the desired date-time and direction filters) and Save as csv or html (from File menu).

For example here are the step-by-step instructions to export all CDR's from a traffic sender:

1. Launch MManage and login
2. Open the CDR form
3. Below the (Re)Load button, select "All" from the drop-down box
4. Select desired date-time interval (change from "Today" to "Custom" and select start/end date)
5. Launch View menu -> Filter -> Set Direction filter and for the source select Type: Traffic Sender, Name: XY and click OK.
6. Click the (Re)Load button on the CDR form
7. Export to file from File menu -> Save as (select html or csv file type) and send the resulted file to your traffic sender if needed

5.26. Server software problem (service unavailable)

1. Restore the last good configuration (Stop the service with stop.bat, copy all files from the lastconfig directory, near the current config and restart the service with start.bat)

5.27. Server OS, Database or Hardware problem (server unavailable)

1. Follow the failover plan.
2. Call the administrator

5.28. How to restart the server service

MManage->Administration->Server Console->Connect and send the „servicerst" command

5.29. How to restart the server box

-MManage->Administration->Server Console->Connect and send the „pcrst” command
-If you cannot connect with MManage, you can find a small program in the vclients directory named „serverrst” (usually at C:/Program Files/VCLIENTS/serverrst.exe
-If these does’nt work, then the server has a serious problem. Follow the failovering plan and call the administrator

5.30. How to restart a GSM gateway

-MManage->Administration->Server Console->Connect and send the „client,XXX” command, where XXX is the gateway name or ip address. When connected to the gateway, send the „pcrestart” command
-if this does not work, then try to connect with remote desktop to the required gateway
-if the gateway is unreachable, then the pc or the internet is down.

5.31. How are the incoming calls from the gsm network handled?

Depending from Gateway Configuration *inccalls* value.
(0=drop,1=hold a little then drop,2=auto forward,3=forward to server as forwardnum,4=forward to number requested by dtmf)
Check the Gateway Configuration for more details.

5.32. Routing test calls to a dedicated gateway

set the calledpriority to the techprefix of the traffic sender
<u>calledpriority:</u> all calls with the specified techprefix will prioritize this gateway (but other techprefixes can go to this gateway also)
<u>testprefix:</u> only the specified techprefix can go to that gateway (but the specified testprefix can go to other gateways also)

so if you want a dedicated gateway for a techprefix, then you have to set the calledpriority and a testprefix too

example:
update tb_users set calledpriority = '987', testprefix = '987' where username = 'TESTGW'
then all calls with techprefix 987 will go to TESTGW with high priority
in case when the TESTGW channels are not available, the calls can be routed to an other gateway

5.33. How to disable PIN request on GSM gateways

<u>The easy way</u>
Set up the pincode entry under the [gateway] or [phoneX] section with the valid pincode. The gateway service will remove the pincodes automatically.
<u>The hard way</u>
1. Start GWTest and switch to the preffered channel/simpos
2. „login” with: **AT+CPIN=xxxx** (where xxxx is the original pin code)
3. Disable pin code request with: **AT+CLCK=”SC”,0,”xxxx”** (where xxxx is the original pin code)

4. in the next switch on, the sim will login to the gsm network automatically

5.34. What are the minimal global settings that must be correct on servers?

On the “Configuration” form select “Basic” settings and check at least the following values:
LocalIP, LocalInternalIP, LocalDomain, currency, Routing, emergencydir, creditunit

5.35. How to add a new traffic sender?

In the “Users and Devices” form select Traffic Sender. Load the list and then hit the “New” button. Then you have the option to clone an already existing traffic sender. Set up the authorization correctly!

5.36. How to add a new sip enduser?

In the “Users and Devices” form select Endusers. Load the list and then hit the “New” button. Then you have the option to clone an already existing traffic sender. Set up the authorization correctly! Check the credit and prepaid/postpaid option!

5.37. How to add a new Mizu VoIP-GSM gateway to the server?

Mizu gateways will register automatically on the server. You may adjust the properties when the gateway is present. After that, you have to set up its sim channels correctly.

5.38. How to add new simcards (sim packet)?

Create a new packet in the “SIM Packets” form. Set up a meaningful name, specify if is postpaid or prepaid and walk through the other options (ownership, access list, recharging options, etc)

5.39. How to add a new simcard?

GSM channels will register automatically on the server. Then you have to set up its properties (to which packet it belongs, recharge options, owner, etc)

5.40. How to set up basic routing?

On the routing form add routing patterns to cover all possibilities (directions and times). Then you have to add your sim packets or other direction in desired priority order. Specify as many simpackets with the same priority as you can (so the server can do the routing too after the other conditions. For example the quality.)

5.41. How to set up basic billing?

On the “Price Setup” form add a new “Invoice and statistics” entry. Then you can add packets to it, which will define the traffic direction when the actual packet will be active and the price.

5.42. Where can I check the logs and traces?

1. “Logs” form
2. “Server Monitor” form
3. Set up your trace level in the “Configurations” form (filer after the “log” expression)

For more details see here and here.

5.43. The conversation volume is too loud. How can I change the volume?

In the Gateway Configuration check the followings: volumein, volumeout, vgr, vgt

5.44. How to register your Mizu Gateway to a H323 gatekeeper?

In the Gateway Configuration check the followings: gkip, gkpassword, gkdiscover, gkprefixesX

5.45. What ports are used in the system?

Standard SIP signaling port: 5060 (TCP and UDP)
SIPS: 5061 (TCP/TLS)
Access port: 80 (TLS)
Secure access port (443 TLS)
Default H323 signaling port: 1720 (TCP)
H323 signaling port used by Mizu gateways: 1721 (TCP)
Rdesktop port: 8836 TCP
SQL Server port: 1433 or 2223 TCP
“Voice Here” port: 44444 UDP
Mizu server admin port: 9885 TCP
Mizu server comm. port: 9886 TCP
Mizu server log port: 9889 TCP
Virtual SIM port: 9886 UDP
H323 additional port: configurable dynamic TCP
Media ports: configurable dynamic UDP
WebServer: 80 TCP
FTP: 21,22 TCP

The exact port numbers used by softswitch can be listed from MManage -> Config menu -> Network - > Active Ports.

5.46. My gateway restarts too often

Check the watchdog settings. For example the gateway will restart if no traffic is routed on it for 3 hour by default. Also check the maxwrongcalls and maxnotconnectedcalls settings

5.47. H323 signaling problems

Check your firewalls.
Check Gateway Configuration: onlyg7x, connectwithmedia, enableh245tuneling, faststart.

5.48. How to set up the automatic credit recharge?

First you have to set up the "Message Rules".
The packet must be set to prepaid. Proper Credit Request/Charge command must be defined. See at 4.6.1. SIM Packets
SIMcard "Credit and Recharge" setting must be set accordingly.

Message Rules types:

-0: msgbgn need to be replaced with msgend before further processing
-1: if msgbgn was found in the sms than it means a successful recharge
-2: credit value between msgbgn and msgend
-4: if msgbgn was found in the sms than it means a failed recharge

5.49. The automatic credit recharge is not working

Check mincreditonrequest, creditrequestival for the packet.
SIMcard "Credit and Recharge" setting must be set accordingly.
Check the CreditRequestFail and CreditChargeFail (the server will try only 5 times. Reset to 0 if the problem is eliminated)
Check the other fields in the simcards regarding to credit charge and request. (fieldnames that contains the "credit" word. You have to check the "All Fields" checkbox on the SIM Channels form to see those fields)
Check if you have charge card for the required simpacket.
Check Logfiles (filter for "credit")

5.50. How to monitor the credit automation?

Method 1: Launch MManage -> Sim Platform -> Credits and check the "credit history" queries
Method 2: Launch MManage -> Monitoring -> Logs and filter for "credit related"

To monitor the credit automation for a selected simcard, you can filter after the simid both in logs and in the Credit form.

5.51. Gateway and channels are inactive

Check if the gateway has internet connection.

5.52. How calls are processed

1. The SETUP or INVITE signal arrives from the traffic sender
2. If the caller is not allowed by the firewall, the call will be silently dropped
3. If the caller is blocked (e.g. DOS attack protection), then call will be silently dropped
4. Caller authorization (by source IP address, username/password, techprefix, etc)
5. Check the call parameters. If doesn't fit into the predefined limits, the call will be dropped (example: too long called number)
6. Rewriting the called number if any Prefix Rule Match
7. Normalizing the called number (validating call prefix)
8. Searching for the best routing pattern
9. Searching for best route direction (available channels, priority order, round-robin, LCR, BRS, failovers, rerouting. etc)
10. Calculating the maximum speech length based on caller credit
11. Checking class 5 features and other endpoint settings (media routing, early-start, etc)
12. Initiating protocol conversion if needed
13. Routing the call to destination
14. Checking for call status, dropping if time exceed and other call monitoring tasks
15. Collecting CDR records at the end of the call
16. Calculating the prices of the call (realtime billing)

5.53. How to set up holiday billing

In the price form in "Time Definitions" select the "Holiday" entry
Set the priority higher in the Directions settings

5.54. How to treat specific weekends as weekdays

Set up a new entry in the holidays form and don't set as holiday (uncheck the checkbox)

5.55. How to add endusers

Endusers can be added to your server multiple ways:

- MManage admin user interface
  - Users and devices form -> click on the New button
  - Users and devices form -> right click on the New button for more options
  - Import users from Config menu -> Users
  - Import data from file menu -> Import/Export
  - Generate users in bulk from the "Generate users and PIN's" menu item
  - Quickly add an enduser: right click on the "Users and devices" tree view node and select "Add enduser"
- API: you can use the API to programmatically add users

- Softphones: users can self register to your service also from the customized softphones
- Web control panel: the VoIP servers ships with a web control panel where your users can manage their settings, recharge or see their call history and statistics. On the first (login) page of this control panel there is a “New user” link which can be used by your new customers to register themselves.
- Database: you can directly connect to the database from any environment or programming language to instert new user records. Example: insert into tb_users (type, username, password, credit) values (0,’username’,’password’,0)
- Web site: you can create endusers also from your website using the VoIP server API or via direct database instert

5.56. SimChange settings from the command line

format:

```
 simchange1= 2004.03.05/13:00:00 - 2004.03.07/13:00:00 - 8936302403070132426 (from date - to date)
   or
 simchange2= 10:20:00 - 10:26:00 - 8936302403070132426 (every day from time to time)
   or
 simchange3= 2/10:20:00 - 7/10:26:00 - 8936302403070132426 (from Tuesday 10:00 to  Sunday 10:00)
   or
 simchange4= 6/00:00:00 - 7/24:00:00 - 8936302403070132426 (Saturday and Sunday)
```

there is a priority order from top to bottom (simchange1, simchange2, etc.) numbering begins from 1 without holes
tip: you can set date-hour prioritization
tip: 24:60 is a wrong time (minutes ends with 59)
tip: on day and exact date settings the roundrobin trick is not working

5.57. How to reenable blacklisted but good numbers

Do the followings to reenable good target numbers:
- In MManage -> direct query, under the misc section check the “reenable blocked but good numbers” section
- delete old number from the helper table (section 0)
- run the query from section 1. this will load blacklisted but good number. The query execution may take 15 minutes
- list found numbers (section 2) and check it agains the blacklist (section 3)
- now you may delete blacklist entryies or set the “sure” level lower. First check the requested blacklist entry agains the query in section 4 (found numbers may be only a subset from the blacklist entry and in this case you may not delete or modify the blacklist. But if the asr and acl values are good for the blacklist entry, you may delete or modify it). Before you delete or modify the blacklist entry, check the comment (why was that number blocked). Number with comment “jukak” or “autdisabled monthly/weekely/daily” should be deleted or changed without problems.

5.58. How are different currencies handled?

In the global configuration, a global currency can be defined by the "currency" setting. For example 'EUR', and there is the possibility to convert other currencies (used for pricelists, simpackets, users) to this "native" currency. For prices defined in "Price List" form, there is a possibilty to convert all input prices in "native" currency by checking the "Convert to XXX" checkbox. In this manner for example you can import a pricelist in other currency and that will be converted automatically in native currency when calculating CDR prices.
The conversions are done based on the settings in the "Currency Converter" form. You should update the conversion rates here as frequently as possible.
If you wish, you can leave the original value intact, so you can make your billing in other currencies than the native.
For every simpacket you can also define the currency, wich will affect the simcard credit calculation (automatic simcredit requests and recharges for prepaid simcards). Simcredits can be converted in the native currency format if the "convertsimcreditcurrency" configuration option is set to true. So you can have simcards in different countries, but all simcredits will be shown in the native currency.
For endusers and traffic senders you can also define different currency format in the Users and Devices form, Billing tab. The currency format defined here will be taken in consideration by the billing process.

Realtime price calculation in CDR records and the credit calculations for prepaid users are always done in the global currency (can be set up in configuration->currency)
However, you are able to set up your pricesettings in any currency. Automatic conversion is done when the given currency is not the same as the "global currency". The conversion is done by predefined rates. You can set these rates in the "Currency convert" form in the MManage.

5.59. How is VAT handled?

You should try to use prices without VAT included all ower in the system (for pricelist and for simcards)
VAT included pricelists can be easily converted to net values by checking the "Convert to NET value" checkbox in the "Price List". You should enter the VAT percent in the "VAT Value" editbox for proper calculations.

For simcards you can setup the VAT value in the Packet options ("VAT" editbox). If you set the "convertsimcredittonovat" global configuration options to true, than sim credits will be automatically converted to net values. For examlpe after an automatic credit request, the credit value in the received messages (SMS) will be automatically conveted to net values.

You should set up the appropiate VAT values for users too, wich will be taken in consideration during the billing process.

5.60. How the check your ASR (or ACD, SL, CDRC) for the traffic sender "A" in the last week.

1. In the date-time drop-down list, select the "Last Week" field
2. In the "Select Direction" form set the "Source" (left side) "Type" to traffic sender, and select "A" in the "Name" drop-down list (or type "A" manually)
3. Launch the "Basic Statisitcs" form under Monitoring.
4. Clear the "Group by" option (select the first "-" line)
5. Make sure the ASR checkbox is checked
6. Click on (Re)Load

7. Depending on current server config and current load this query may take some time (on a usual configuration this will take 2 second)

5.61. How to add endusers (basic settings)

1. Go to MManage -> Users and device form, and select enduser type
1. Select an already existing user wich has the same caracteristics as the required new endusers
2. Hit "New User" and than accept the the copy from existing option (cloning)
3. Check at least the following fields: username, password, parent id, authorizaton type (usually username/password), prepaid/postpaid, billed user
4. Check other settings
5. Save

5.62. Basic callcenter tasks

1. Setup your server as for a normal sofswitch (routes)
2. Create campaigns
3. Add callcenter operators
4. Assign operators to campaigns
5. Add or import clients
6. Assign clients to campaigns
7. Add presentation locations
8. Setup global callcenter configurations
9. Operators now are ready to start there MAgent application
10. Check statistics
11. Print invitations
12. Use checklist when you are on presentations

5.63. SRTP

SRTP media encryption is fully supported by the VoIP server and it can be configured globally or per user/device.
The SRTP module can be enabled/disabled from MManage -> Configuration wizard -> Roles and features page and its default behavior can be set from the Network page.
You can also configure the same with the **srtp** global config from the Configurations form.
Possible values:

- -2: IGNORE: disable all srtp processing (will ignore all srtp related code).
  Will completely disabling the SRTP module.
- -1: AUTO: like 1/FORWARD, but like NO if target is PSTN number.
  Default. Will auto guess the best behavior.

- 0: DISABLE: disable SRTP (don't forward SRTP and respond in 200 Ok without srtp in the sdp).
  Can be used to remove SRTP without any extra server side processing if the client allows plain RTP answer for SRTP initiated calls.
- 1: FORWARD: forward as-is when possible (try to bypass and failback to local decode if the peer doesn't accept SRTP).
  With this option you can usually avoid SRTP related transcoding.
- 2: PROCESS: if incoming call is with SRTP, then transcode locally and forward as simple RTP. don't forward SRTP for outgoing calls.
  You can set to this value to let the server process (decode/encode) the SRTP and always forward unencrypted RTP.
- 3: YES: offer SRTP for outgoing calls (even if incoming was without SRTP). For incoming calls this will act like AUTO.
  You can set this value to transcode raw RTP to SRTP.
- 4 FORCE: always offer and reject call if not with srtp.
  Same as YES, but will allow only SRTP calls also for inbound.

Default value is -1/AUTO

There are a few extra SRTP related global configurations which you can change from the “Configurations” from by searching for “srtp”. The defaults are optimal.
srtp_auth: -1: yes/default, 0: no, 1+: specific algorithm
def_srtp_alg: defaults to AES_CM_128_HMAC_SHA1_80
The followings are supported: AES_CM_128_HMAC_SHA1_80, AES_CM_128_HMAC_SHA1_32, AES_CM_256_HMAC_SHA1_80, AES_CM_256_HMAC_SHA1_32

You can also configure SRTP per user form the Users and devices form -> Functions page -> Media relay section.
This will overwrite the global setting.
(This will set the tb_users.srtp field but the value will be 2 points higher then for the global srtp settings. For example tb_users.srtp 3 means FORWARD)

It is recommended to use TLS transport for the signaling if you need SRTP encryption for the media.

Note:

- *Depending on the circumstances (routertp and whether any local RTP processing is required), the server might just bypass the SRTP stream (P2P media between the peers), it might forward the SRTP stream but doesn’t encode/decode or it might forward with local encoding/decoding. If local processing is required, can be specified with the rtpmightlocalprocess global config. Default is -1 (auto), 0 means never (usually meaningless), 1 means always (which means that the SRTP streams will be always decoded and then re-encoded if required)*
- *If the target party rejects the call due to SRTP or missing SRTP (for example with SIP/2.0 488 Not Acceptable Here), then by default the server can retry the call with SRTP transcoding (with or without SRTP).*
  This behavior can be disabled/enabled/forced with the autosrtp global setting:
  - *0: disable. never auto failback to SRTP transcode*

- *1: enable with auto guess (default value)*
- *2: force. always auto failback to SRTP transcode (with the risk that codec transcoding would be needed instead)*

5.64. Codecs

The sofswitch can route any audio, video and fax codecs.
If the audio stream needs to be manipulated (conference, call recording), the following codec's are supported:
G.711, OPUS, G.729, G.723, GSM, iLBC, Speex, G.722
(Voice recording, "voice here" and conferencing feature will work only with these codecs.)

5.65. How to reserve GSM capacity for a certain protocoll

In the global configuration you have to set the following keys:
reserveforh323: number of channels to be reserved for traffic received from H323;
reserveforsip: number of channels to be reserved for traffic received from SIP.

On packet and gateway level you can specify the maximum number of reserver channels from that packet by setting the "maxalloc" field (defaults to 0!).

When set properly, the server will reallocate the requested best quality channels for the requested direction in every hour.

To manually reserve capacity for sip set the reserverfor field in tb_sims to 1 on the required channels.

5.66. PDF creation in MManage

In order to use "PDFCreator" to save the invoice in PDF, some modifications are needed to the program

<u>1. Make sure that the default PDF printer is set to "PDFCreator"</u>
(MManage -> Menu -> Settings -> Options -> Default PDF Printer option)

<u>2. Open the printer "Options" Form</u>
This can be achieved
-when you save invoices as PDF first time
or
-from Start Menu -> Programs -> PDF Creator -> PDF Creator -> Printer Menu -> Options

<u>3. Setup Auto-Save</u>

Select the Programs->AutoSave form

-check the “Use Auto-save” checkbox
-In the filename template type: <Title>
-Check the “Use this directory for auto-save” checkbox, and select the folder where the invoices will be saved
>Now your invoices will be saved automatically in the selected folder when you select PDF as output.

If PDF creation doesn’t work:
1. check if the “Print Spooler” service is running (set it to start automatically in StartMenu->Settings->ControlPanel->Administrative Tools->Services)
2. reinstall “PDFCreator” by starting the “PDFCreator.exe” from your MManage directory

If you cannot use the “PDFCreator” program, you can change the PDF creator tool in MManage -> Menu -> Settings -> Options -> Default PDF Printer option

5.67. How to setup a new virtual server instance

1. create a new directory named “virtserverX”
2. copy all files from an old virtserver directory (except the logfiles)
3. rename virtserverOLD.exe to virtserverNEW.exe
4. register the new virtserverX.exe as a system service
5. rewrite the start.bat and stop.bat
6. allow the new service (and other executables if any) in the OS firewall
7. create a new ftp directory for the recorded voices files under the “voice” directory and set the correct rights
8. set the database connection settings in the inifile
9. clone an old database instance
10. clean old database tables (MManage ->Advanced->Clean Database Tables)
11. setup default tables (MManage ->Advanced->Setup Default Virtserver Database)
12. upgrade to the newest version (MManage ->Advanced->CC SW Version Migration)
13. setup configuration (MManage ->Settings-> Quick Setup)
14. check port collisions (rewrite portnumbers)
15. configure routing in the mainserver (in and out). Add the virtserver user as traffic sender and sipproxy
16. allocate numbers in the mainserver for the new virtserver instance (callback numbers)
17. start the new virtual server
18. make a testcall
19. check the logfile for errors. test MAgent. check MManage

5.68. Fax detection

Usually you can filter out fax calls by setting the “block711” global setting.

5.68. Handling dynamic & private ip/port

For sip proxyes, endusers and gsm gateways the server will handle ip and port changes automatically. For sip proxyes and endusers the ip/port/transip/transport settings will be changed dynamically only if the “Can Dial” option is enabled for the device.

5.69. Fax settings

faxhost: email-fax gateway domain name or ip
faxuser: smtp username
faxsubject: email subject
faxfromaddr: from field
faxfromname: from name field
faxnormalize:
0 = leave original number
1=normalize
2=normalize, no iec
3=normalize, with iec

faxsuffix: append after faxnumber (usually @host.domain)

5.70. License limitations

You can configure the limits in the “licensecfg” global config section, but the configured values cannot exceed the builtin hardlimits. Most of builtin hardlimits are listed in the “license” section or you can check them in MManage->help->licensing menu. The -1 value means “no limit”.
When the software has no builtin and configured limits, than only the hardware can restrict its performance and throughput.

The following hard **limits** are built in the software:
MAXALLUSERS: to restrict the total number of users and devices (including endusers,gateways, etc)
Devices behind this limit will be disabled (calls blocked)
MAXSIPUSERS: to restrict the total number of sip endusers.
Users behind this limit will be disabled (calls blocked)
MAXGATEWAYS: to restrict the total number of sip, h323 and gsm gateways, gatekeepers and proxies.
Gateways behind this limit will be disabled (calls blocked)
MAXTRAFFICSENDERS: to restrict the total number of sip traffic senders
Traffic senders behind this limit will be disabled (calls blocked)
MAXCHANNELS: to restrict the total number of gsm channels (active simcards)

Server will not accept more simcards to be active, thus no calls will be routed to them.
MAXCCALS: to log exceeding traffic. When the number of the simultaneous call count exceed this limit, a log entry will be written with the following text: “WARNING, max concurrent calls reached (license) [num of calls]”.
MAXCALLSBLOCK : call behind this limit will be dropped with “Max concurrent calls reached” reason code.
MAXCCALSPERMIN: max number of call setup/minute.
MAXSL: max speachlength/day in minutes. All calls will be dropped after this limit is reached.
MAXSESSIONSPEECHLEN: max speech length for all calls in miliseconds.
MAXREGISTRATIONS: max number of registered users (by the nature of sip registration process, this limit can’t be 100% precise)
MAXTAGENTS: max number of maximum MAgent application registrations. There is no way to control wich of the MAgent application will be blocked. (The registration request will be refused when there are more TAgents online than this limit, where “online” means at least registered status in the last 15 minutes)

The following **modules** can be added/removed:

- callcenterout
- callcenterin
- extra
- filtering
- failover
- alerting
- recharge
- gsmextra
- sip
- h323
- simplatform
- simbank

To check for more limitations, search the config with the “license” and “max” keyword.

5.71. Redirect or forward sessions to other domains

Routing to other domain can be restricted by the “fwdtootherdomains” global config setting. This means SIP calls with other domains in the request uri (not listed in the “domainnames” values, and no IP address match)
The following values are defined:
0=don’t forward calls to other domains
1=check if our numbers first (local user)
2=don’t forward mobile numbers

3=forward all required sessions
4=forward all required with “Moved Temporarily”
5=unconditional move (will move all required traffic without checking the caller)

You can forward calls to other domains before any routing configuration will be checked. (with 302 Moved Temporarily)
The following config values will be applied:
Forwardpx: comma separated prefixes to be forwarded. ‘*’ means all traffic (can be used for **failovering and load balancing**). Set to emty to disable forwarding.
Forwardto: IP address or domain name where these prefixes will be forwarded.

This feature is usually not enabled for service providers (if you provide services for costs), and usually is enabled for home or company users.

5.72. Delete old database backup

You can automate backup cleanup by setting the following global config values:
Deldbbackup: days to keep (-1 disables cleenups)
Dbbackupdir: database backup directory
dbdelbackupdir1, dbdelbackupdir1, dbdelbackupdir3: database backup subdirectories
This feature is useful, when the database engine doesn’t have cleanup feature.

5.73. Sales commissions

The sales cost will be calculated to the CDR record “costsales” field. So you can build any statistics based on this value.
Every user can have the “Added By” field filled, wich will point to the sales person where the user belongs.
You can define the sales commission explicitly in the “Price setup” form (Type must be set to “Sales cost”)
If no such price is defined, than the sales commission will be loaded from the actual sales user setting.
The defined commission percent will be calculated for the profit **or** for the enduserprice. This behavior can be set by the “salescomissionfromprofit” global config.
Sales can give up for the users some reduction wich will be substracted from their cost. This can be configured with the “reduction” value for the individual endusers.

5.74. Multihomed setup

UDP ports must be binded to different ip interface (at least the default sip port 5060), or set to different values.

TCP ports must be setup differently
On the gsm gateway set the serverip to the binded ip and the cmdport to the gsmclientport
Change the pipename setting in the gatekeeper.ini

5.75. How to scan your SP for live numbers

The scanner thread can be used to initiate phone calls to a range of phone numbers and will store the validity status of these number in CDR records.

Phone numbers can be specified by a range (scan_from, scan_untill) or loaded from a text file (scan_filepath).
The speed of the thread can be controlled by the following options: scan_callival, scan_callsperminute, scan_maxcalls (you might specify all these 3 constrains for a better control).
If there is no error in the configuration, the scanner thread will start immediately withing 2 minutes after service startup.
Otherwise the scanner can be started/stopped with the "checkscanner" console command (will start if the parameters are ok and scan_enable

At the end of the scan, you can query the results from tb_cdrs.

The following global config values must be set accordingly:
scan_enable –set to 0 to disable scanning or 1 to enable (The scanner thread will start automatically if this is set to 1 and will stop automatically on 0 value)
scan_from –range start
scan_untill –range end
scan_filepath -to load the numbers from file (scan_from and scan_untill will not be used in this case)
scan_callernum –a valid username (phone number) from "users and devices". Proper routing must be set for this caller. (A number)
scan_callival –new call attempt interval in msec. (delay between 2 calls)
scan_callsperminute –max calls/minute to initiate
scan_maxcalls –max simoultan calls
scan_stop_at –at what stage the call setup progress will stop.
    0=trying,1=progress (default),2=ringing,3=connect
**all global config can be listed if you filter for the "scan" word"*

The results can be seen in the marker field of the cdr records:
    SU1=unknown
    SS1=trying (scan success 1: the last status was: trying) -2
    SS2=progress (scan success 2: the last status was: progress) -3
    SS3=ringing (scan success 3: the last status was: ringing) -4
    SS4=connect (scan success 4: the last status was: connect) -5
    SF1=disc code (scan failed 1: the last status was: disc code received) -6

SF2=bye (scan failed 2: the last status was: bye received) -7

For example the following query will list all live numbers when the scan_stop_at is set to 1:

```
select callednumber from tb_cdrs with(nolock) where datum > X and callerid = Y and marker in ('SS2', 'SS3','SS4')
```

You might also consider a number to be live on special disconnect reason codes (like user busy)
Disconnect codes are stored in tb_reasoncodes (there are mappings to SIP codes)
For example the following statement will list all numbers found as valid in the last 7 days including calls with “Busy here” disc reason.

```
select callednumber from tb_cdrs with(nolock)  where datum > GETDATE() - 7 and (marker in ('SS2', 'SS3','SS4') or
(marker = 'SF1' and discreason in (4486)))
```

5.76. Routing calls from virtual servers

The actual virtsever must be configured as a traffic-sender in the mainserver.
In the virtserver, the mainserver must be configured as a sip-proxy.
Setup the routing in the virtual server and in the mainserver too.

5.77. Routing calls to virtual servers

From the mainserver there are two possibilities:

-set some phone numbers “location” to virtserver id and /or
-setup the routing (for a range of numbers defined by a prefix)
The actual virtsever must be configured as a sip-proxy in the mainserver.

In the virtserver:

-setup the callbacknumber ant it’s routing correctly in global configuration and /or
-assign real phonenumbers for operators (or endusers)
In the virtserver, the mainserver must be configured as a traffic-sender..

5.78. How to add plugins in MManage

Create your custom exe.
In the MizuManage.ini under the “plugins” section add your plugin entry:

```
nameX=MyPlugin
cmdX=MyPlugin.exe
```

X is a number starting from 0 to MAXPLUGINCOUNT

The MManage will pass the database access parameters in the command line.
Any other parameter must be readed from the inifile.

Command Line: dbserverip,dbport dbname username password connectionstring

**Please note that appserverip may differ from dbserverip!*

Direction parameters will be located under the “parameters” section. Read them as soon as possibile.
The following keys are defined:

- fromdate, todate
- source, destination
- fromtype,fromid, frompacket, fromsimid, fromgroup, fromcamp
- totype, toid, topacket, tosimid, togroup
- fromname, frompattern
- toname, topattern
- fromenabled, toenabled
- fromidlist; toidlist
- fromsimidstr; tosimidstr
- fromip, toip
- fromsimgrouplist, fromusergrouplist
- tosimgrouplist, tousergrouplist
- reloadgroups reloadgroups

5.79. Request/response target address

Because the server will try to send the sip messages to all possibile addresses, sometime it will misroute it.
With the **denyaddr** setting you can restrict the address posibilities. It can be set global or user level.
The following values are defined:

- NULL *//will load the default denysetting*
- * *//allow all*
- loopbackip, loopbackaddr *//127.0…, 0…*
- localip, localaddr *//locally configured or autodetected ip/port*
- lanip, lanaddr *//192.168, 10.0, etc*
- privateip, privateaddr *//local + lan*
- notfromip, notfromaddr *//the address from where messages are received*
- notsigip, notsigaddr *//the address sent in sip signaling*
- specificip, specificaddr *//any ip or ip:port*

Examples:

to restrict the target address for a sipuser you don’t allow local and lan ip: **privateip**
to restrict the target address for a virtserver you don’t allow: **localaddr**
to restrict a misconfigured proxy wich is always telling you 11.12.13.14: **privateip, 11.12.13.14**
on a virtserver you must deny only the **localaddr** *for the mainserver (wich is configured usually as a sipproxy) and for the operators you can deny* **privateip**

5.80. Extra charges

To increase the call durations (and the price) for the CDR records, set the “incduration” global config option to true and then set the “incduration” field for the users to one of the following values:
0 = no increase
1 = small inc ~ 1%
2 - normal inc ~ 2%
3=bigger inc ~ 3%
4=very big inc ~ 4%
5=very big inc ~ 5%
6=abnormal :)
above 6: means a fix percent increase

You might also set the “incdurationmode” global config option to 1 if you wish the increases to be applied only for endusers and not the resellers.

5.81. Automatic prepaid credit expirity

You can set prepaid credit by the “Add with elapse” button to elapse automatically.
The following configurations are defined:
Creditunit: How much credit means 1 day
Maxcreditelapsedays: max number of days when the credit will elapse
Accelapsedays: the number of days from creditelapsedt when the account will expire
Tb_users. Creditelapsedt: date-time when the credit will be expired
Tb_users.Accelapsedt: date-time when the account will be expired

5.82. Simple prefix rewrite

To rewrite prefixes on router number normalization, you have to set the following global config values:

    prefixrewritestr: the original prefix
    prefixrewritefrom: keep from
    prefixrewriteto: inserted string

for example to handle the hungarian roaming prefix: 08 + SK + BK + NSN +SN you have to set the following values:

    prefixrewritestr: 08X…
    prefixrewritefrom: 9
    prefixrewriteto: 36

5.83. Short number and internal billing

You can set the short number billing mode by the “shortnumbill_type” config value. When set to 0, all short numbers are billed with the price set in “internal_providercost” and “internal_endusercost”. When set to 1 (default),, then the “999” prefix is inserted before the called numbers, and you have to create billing entries for these numbers.
Numbers are treated as “short” if its length doesn’t exceed the value set by the “billshortnumlength” global config option.
Calls between endusers are billed with the “internal_providercost” and “internal_endusercost” values (defaults to 0).

5.84. Cropping sound on ring when playing voice (ringtone, announcement or any other prompt

Cropping sound on ring when playing voice (ringtone, announcement or any other prompt)
Set the stopplayeronrtprec global config value.

5.85. How to play user credit in IVR

Create a “Play number” action.
Enter [credit] for the file name.

5.86. Mizu Server security

Moved to “Security and account limiting” section

5.87. Ringtone for IVR forwarded calls

Can be controlled by the playivrfwdringtone global configuration
We have 4 options here:
0=no ringtone from the server (instead we can play any file directly from IVR). In this case if there is no other IVR file playback in progress, the client can hear the ringtone generated by the called endpoint
1=generate ringtone on ring received if there are no other IVR playback in progress (in this case the client will hear ringtone even if the called endpoint is not generated –i.e. it just sends ringing message in signaling). This is the default setting.
2= generate ringtone on ring received even if other file playback is in progress (stopping the old playback)
3=generate fake ringtone immediately after call was sent to routing

5.88. How to enable codec transcoding

By default the server will try to auto-negotiate the "best" possible common codec between peers and it can automatically failback to transcoding if there is no common codec, thus most probably you don't need to touch the code or transcoding related settings.

In case if you wish to prefer or force some codec for a SIP endpoint, then just set the "Codec" setting from the "Users and devices" form "Functions" tab (this can be set for Enduser, Traffic sender and SIP Server entries). *This will set the tb_users.choosecodecs field accordingly.*
In case if you wish to force transcoding, then also select the "Transcode" checkbox. This should be done only if you know that the caller and called pary will not have a common codec. *This will set the tb_users. needcodecconversion field to 1.*

You might also need to adjust the convertcodecs global config after your needs (list all the possible payloads).

More details:
*If one of your peers has limited codec capabilities or the accepted codec(s) doesn't match with the sender codec's, then set its "**needcodecconversion**" to 1 (for the target user which is usually a SIP Server user or Enduser)*
*Set the "**convertcodecs**" global config value to the target codec payload list. The default value is 0,8,18,3,105 which means PCMU,PCMA,G.729,GSM and OPUS-wideband. The list should include both the caller and called party codec's.*
The server is able to transcode between the following codecs: G.711 A-law , G.711 A-law ,G.729, G.723.1, GSM, Speex 2,3,4,5,6 (narrowband, wideband and ultrawideband), OPUS (narrowband, wideband and ultrawideband), G.726 and G.722. You might also have to set the "choosecodecs" field for the target user (same as the "convertcodecs" global config value) and the "convertcodecsforced" global config option to true.
Be aware that codec transcoding require a high amount of CPU usage. For example one CPU (core) can handle around maximum 10-200 simultaneous transcoding between PCMU and G729 on full load (depending on your CPU type).

5.89. Possible NATs and firewalls

There are several ways UDP might be handled by a specific NAT or firewall implementations, these are categorized into:

Full Cone NAT
A full cone NAT is a solution, where all requests from the same internal IP address and port are mapped to the same external IP address and port. Furthermore, any external host can send a packet to the internal host, by sending a packet to the mapped external address.

Restricted Cone:
A restricted cone NAT is a solution, where all requests from the same internal IP address and port are mapped to the same external IP address and port. Unlike a full cone NAT, an external host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.

Port Restricted Cone
A port restricted cone NAT is like a restricted cone NAT, but the restriction includes port numbers.
Specifically, an external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.

Symmetric Nat
A symmetric NAT is a solution, where all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port. If the same host sends a packet with the same source address and port, but to a different destination, a different mapping is used. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.

5.90. How to disable CLI for all outgoing calls

For the sip proxy users(s) in the rewriteanumber put "anonymous". Make sure you don't have identity forward and identity rewrite (both of them must be empty)

5.91. How to change the database username/password

The auto installer will use the "sa" username with "srEgtknj34f" password by default.

Install the SQL Server Management Studio if not already installed.
Login with the old credentials.
Open "Security" -> "Logins". From here you can easily add/remove any users or change passwords.

Before changing the default user settings, stop the mserver service (By launching the stop.bat file or from Services).
Change the database settings in the mizuserver.ini configuration file.
Restart the voip service.

5.92. Performance optimizations -fine-tune

The Mizu VoIP server is optimized by default and will also fine-tune itself at runtime.
Do not change these default settings unless you have a solid reason to do so on high traffic servers if you have performance issues.

Use the v_check_blakclist_int sp if there are a lot of blacklisted numbers (global config blacklisttype = 2)
Disable rtp routing whenever possible (set the default to "route rtp if both behind nat")

Set getpacketloss global config option to 0 (this will enable fix rtp relay)
Disable resellerbilling if possible (Set to 0 if you are not working with resellers)
Set trace level to 1
Switch off dbmaint_mssqldbbackup (set to 0) and set db backup from ms sql studio
Set outboundprefix1, outboundprefix1 and outboundprefix2
Disable early media on client and server side
Disable retrybilling, retrycalldir and retryrouting
Increase media frame/packet
Set uniqueportperuser to 0
Set “allowlastwrongnum” to false
Set “usesockrecmutex” to false
Set “usemencmutex” to false
Set “rtpalwaysroutebyfs” to 2
Set “fs_timer” to 2
Set “dbcache” to 2 if you have enough RAM
Set “checkrtpevents” to false if you don’t need DTMF based conference or transfer
Set lookupcdrcalldirection to false (disable direction name in cdrs)
Increase maxasyncrecthreads (set to number of available cpu)
Set threadsafeoops to 1 (or to 2 if there are stability issues)
Enable/disable useasynctransportthread (global config option for async send, defaults to -1 which depends on cpu count); set the useasynctransportthread global config option to 1 if too high load for the main thread
Use MS-SQL query governor
Turn off disk write-cache buffer caching if you have a separate power supply
Set the usevgetpriceint to true (also requires dbversion 124; with this mode, + and 0 prefixes are not allowed in the billing. Only numbers)
Setup user caching (set “cacheregistrations” to 2)
Set checkfilecontents to false.
Set updateusrcalldate to 0
set #define MAXMSGLEN to 2000
Adjust DEFQUEUELEN
Set REBUILDREGCLIENTS to -1
Remove performance counters, lasttoops
Set offlinechat to 0.
Set checklocalnumbers to 0
Set blacklisttype to 2 if you have a lot’s of blacklisted numbers (convert to tb_blacklist_int)
Set quickanswerforoptionnotify to 2
Set simultancallscreditcut to 0 or 1
Finetune opus settings (lower bandwidth)

Set resolvedns to 1 (for non blocking / only delayed resolve)
Set useencryption to 0 if no mizu encrypted clients are used
Set checkiostcpfastencrypt to 0 if no iOS encryption is used
Turn off cpu affinitiy by settin the followings to 0: cfg_cpuaffinity, autocpuaffinity, autocpuaffinitysubprocess, autocpuaffinitythreead
Set usemencmutex to false (migt result in instability)
Set waitforsqlupdate to 0
Set reusedblockedep to 2
Set validatesipmessages to 0
Set "useasynctransportthread" to 1 (especially if TCP is used)
Optimize the number of threads ("thead"settings)
Increase REGISTERIVAL,upperexpire, userofflinemin, upperexpiremin
Set allowforkforsignaling to 0
Set usertpmutex to false
Set allowlogloss to 2
Increase cfg_sipmsgresendival
Turn off creditcheckforpostp
Turn off allowforkforsignaling
Turn off creditcheckforts
Turn off lookupcdrcalldirection
Turn off brs if not needed. ("lcr_brs")
Turn off the dynamic firewall.
Turn off client credit (balance) notifications (set sendusercredit to 0)
Turn off rtp mutex (set "usertpmutex" to false)
Disable RTCP (set enablertcp to 0)
Remove rules if not needed (checkprefixroules)
Turn off checkdbconf or set to 1 if db interface needed.
Turn off checkepvalidity (Set to 0)
Turn off storecdrcomments
Turn of A number authentication if not needed (anumberhandling=0)
Turn on quickrtp
Turn off forking (media and signaling also)
Turn off rerouting
Set fastackforcalls to true
Turn off input validation (validateinput)
Turn off A number lookup for users when not needed
Increase the maxregelivetime on high CPU load
Decrease the maxregelivetime on high memory usage

Increase or turn off keep alive message interval
Use only UDP transport
Set “alwayssethttpanswerport” to false
Set “maxhttpbuffering” to 0
Set “checkwithzerocredit” to 0
Set cdrextra to 0 and remove the following fields from cdrs: billingstep,unitprice,billingentry,origduration,costenduseru
System: SQL optimizations (tuning wizard) and profiling according to your workload.
From v_getroutingtime remove @called_norm2 LIKE tb_routing.calledprefix +'%' and use fixed length matches only
Don't use LCR
Check your network card settings:
-disable flow control
-enable interrupt moderation
-enable IP/UDP checksum calculation offload if you have a performant network card / less CPU inverse or disable if your CPU is high-end bug the network card is moderate only (some network cards might not be able to keep up with the checksum calculation workload)
Run SQL profiler to find out low performing queries
Adjust MAXDOP option (to prevent one query stealing all your cpu time)
Disable all software firewalls (disable the windows built-in firewall). Firewalls has a big impact on UDP traffic.
Set ipdontfragment to 1 (test if peers can still receive large udp packets such as SIP INVITE)
Add the app folder to your virus scanner (defender) exception list
-Some other OS optimisations:

- Disable QoS Packet Scheduler
- Disable TCP/IP v6
- Disable Indexing
- Set screen saver to none
- Sound Off and Sound scheme to No Sounds
- Adjust Visual Effects to best performance
- Clear C:\Users\Administrator\AppData\Local\Temp folder
- Go to the cmd prompt and type powercfg.exe -h off to disable hibernation. (This also deletes the hiberfile.sys from c:\)
- Power Options (from Control Panel) Change option to High Performance (http://support.microsoft.com/kb/2207548)
- Configure SNMP
- Defragment the C drive

-RTP routing offload

-From server management install the “Network Policy and Access Service” Role, “Routing” feature
-From “Routing and remote access” right click on the server node and select “Configure and enable Routing”
-Select the “Custom” option and then select “NAT”
-Rigth click on the “NAT” node and from “New Interface” select your network card connected to the public network/internet
-Select “Public” interface and select “Enable NAT on this interface” then click OK to save

-for the mserver global config, set the followings:

-rtpoffload = 1

-for the “rtpoffloadapter” enter the name of your network adapter. (default is: “Local Area Connection”)

-Increase page file maximum size to allow more virtual memory (If auto managed then Windows might limit it to 4 GB)

-System/MS SQL optimisations

-Set forced parametrization: ALTER DATABASE mserver SET PARAMETERIZATION FORCED

-optimize indexes and fillfactor: https://www.brentozar.com/archive/2013/04/five-things-about-fillfactor/

-set the v_selpattern to return only top 10

-remove unneeded called prefix length checks from v_getroutingtime (PATINDEX ('%,'+substring(@called_norm2,1,1)+',%', ','+tb_routing.calledprefix+',') >0)

-increase the recovery interval (to 5 or 10 depending on your needs)

-optimize for ad hoc workloads:

```
sp_configure 'show advanced options',1
reconfigure
go
sp_configure 'optimize for ad hoc workloads',1
reconfigure
go
```

-create 8 tempdb with equal size

-add TOP X for v_getroutingtime (1 is enough if you are not using failovering between patters. Otherwise set to a reasonable value)

-review the following settings

```
SetCfgStr("supervisor","canrestartformalfunctions","1");
SetCfgStr("settings","mincdrcount20min","1");
SetCfgStr("settings","maxmemoryutilization","800000");
SetCfgStr("settings","minmemoryutilization","280000");
SetCfgStr("settings","maxgkmemoryutilization","90000");
SetCfgStr("settings","priorityboost","false");
SetCfgStr("settings","wrongnumcache","100");
SetCfgStr("SIPSettings","MAXSUBSMSGCOUNT","35000");
SetCfgStr("SIPSettings","MAXWRONGMSGALLOWED","4000");
SetCfgStr("SIPSettings","MAXWRONGAUTHFROMIP","3000");
SetCfgStr("SIPSettings","MAXFAILEDAUTHENABLEDIPPORT","170");
SetCfgStr("supervisor","maxnocdrmin","6000");
SetCfgStr("supervisor","maxnologival","20");
SetCfgStr("SIPSettings","MAXH323GKCDRCACHE","400");
SetCfgStr("CallCenter","maxcallatonce","1000");
SetCfgStr("CallCenter","maxcallsperminute","5000");
SetCfgStr("CallCenter","predectivemaxsucccalls","500");
SetCfgStr("CallCenter","stopwrongcdrc","400");
SetCfgStr("settings","allowlastwrongnum","true");
```

```
    SetCfgStr("settings","maxroutereqpermin","5000");
    SetCfgStr("settings","MAXEPCOUNTTRESHOLD","1000");
    SetCfgStr("settings","checkmaxcalls","500");
    SetCfgStr("settings","maxauththreads","9");
    SetCfgStr("settings","maxivrthreads","9");
    SetCfgStr("settings","maxasyncqueryhreads","9");
    SetCfgStr("settings","maxroutingthreads","9");
```

-enable Disk Write Caching (only on stable servers with backups)
-use SSD and as many RAM as possible
-If you are using SQL Server Enterprise and high disk usage and low CPU usage: Use Compression
-put the main database, tempdb and logs to deparate disks
-put the tempdb to a ramdisk if you have plently of RAM
-use in-memory tables
-increase the 'min memory per query' option if your server has a lot of memory available
-set the 'priority boost' SQL Server options to 1 of you have a dedicated sql server
-split the database to multiple files
-on servers hosting MSSQL, format disk with NTFS 64K allocation unit size
-fine tune MSSQL based on your hardware/configuration/requirements:

```
        USE mserver;
        GO
        EXEC
        sp_configure 'cost threshold for parallelism', 30;  //set to 20-90 (50)
        sp_configure 'max degree of parallelism', 4;  //set to number of processors / 4
        sp_configure 'blocked process threshold', 60 ;
        sp_configure 'optimize for ad hoc workloads',1  //might degrade the performance!
        sp_configure 'query wait', 60;
        sp_configure 'max server memory', 4096;   //MB  auto set  Max Server Memory to 90% of the system (leave at least 10% or 2 GB for the OS)
        sp_configure 'backup compression default', 1;
        RECONFIGURE WITH OVERRIDE ;
        GO
```

```
SELECT name, value, value_in_use, [description]  FROM sys.configurations ORDER BY name;
```

You can set the 'priority boost' SQL Server options to 1. ...if server is dedicated for mssql
Set OS power management to "High performance"

In case if you are hosting multiple services on your server, you might need to increase the heap size.

Set HKLM\System\CurrentControlSet\Services\AFD\Parameters\FastSendDatagramThreshold value to 1450 (decimal)
Increase HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\LanmanServer\Parameters\IRPStackSize

netsh int ipv4 set dynamicport tcp start=40000 num=64000
netsh int tcp set global dca=enabled
netsh int tcp set global netdma=enabled
Set the power plan to “High performance”
check perfmon /report

https://msdn.microsoft.com/en-us/library/windows/hardware/dn529133(v=vs.85).aspx
http://msdn.microsoft.com/en-us/library/cc558565%28v=bts.10%29.aspx
https://kbsupport.blogspot.ro/2012/10/registry-settings-to-improve-network.html
http://blogs.technet.com/b/vipulshah/archive/2006/11/30/understanding-perfmon-counters-while-troubleshooting-sql-server-performance-issues.aspx
http://support.microsoft.com/kb/126962
http://technet.microsoft.com/en-us/library/cc966401.aspx
http://technet.microsoft.com/en-us/library/cc966420.aspx
http://msdn.microsoft.com/en-us/library/cc558565%28v=bts.10%29.aspx

```
SELECT * FROM sys.dm_os_performance_counters
```

https://forums.spybot.info/showthread.php?4723-Solve-Security-Issues-in-Windows-XP

Increase softswitch stability (and decrease performance) with these settings:

usemencmutex=true
usertpmutex=true
usesockrecmutex=true
threadsafeoops=2
other threading and mutex related
checkepvalidity = 3
recompile with the following settings for TSQueue:

```
#define KQUEUE_EXTRAMUTEX
#define KQUEUE_USECONDITIONVARIABLE 2
#define KQUEUE_USEWAITONADDRESS
```

also check the performance related settings above

NIC performance:

https://blog.serverfault.com/2011/03/23/performance-tuning-intel-nics/
https://docs.microsoft.com/en-us/biztalk/technical-guides/general-guidelines-for-improving-network-performance
https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics
https://helpdesk.flexradio.com/hc/en-us/articles/202118518-Optimizing-Ethernet-Adapter-Settings-for-Maximum-Performance
https://www.tradingtechnologies.com/xtrader-help/tt-system-requirements/appendix-setting-descriptions/nic-settings/

If you are using VMWare, increase the interrupt rate as described here: https://kb.vmware.com/s/article/2040065

5.93. How to reset a failovered gateway/direction

-set tb_users.nopriority to any date-time in the past
-open failoweing and reset the failovered gateways
-check if all gateways have provider pricing

5.94. How to remove the 3 digit techprefix system wide setting

This might be required for old server versions (before 2009) to convert to the latest database format:
Run the following queries and reload the server config.

```
update tb_cdrs set callednumber = substring(callednumber,4,99) where calledid > 0 and LEN(callednumber) > 3
insert into tb_settings (location,filename,inisection,keystr,valstr) values ('global','vserver','settings','dbformat','86174')
insert into tb_settings (location,filename,inisection,keystr,valstr) values ('global','vserver','settings','usetechprefix','false')
update tb_settings  set valstr = 'false' where keystr = 'usetechprefix' and inisection = 'settings'

update tb_users set ccenddate = getdate()+9999
update tb_users set ccstartdate = getdate()+2 where ccstartdate > getdate()
```

To enable variable length techprefix for the authentication, do the followings:
-set the “techprefixlenauth” global config option to -1
-upgrade to the latest v_checkuser stored procedure

5.95. How to enable H.323 modules?

H.323 is usable if your license contains these modules and you have installed the H.323 modules during the initial setup.
Open the global configuration form.
Search for h323
Set the followings to true:
-CAN_hash323
-CAN_runsipproxy

Enable the sip2h323caller and sip2h323 users.

Restart the VoIP service.

5.96. How to add SMS provider(s)?

Content moved to SMS chapter.

5.97. Working with groups using direct SQL

Open “Direcy Query” from MizuManage or MS SQL Management Studio.

```sql
--check related tables
select top 1000 * from tb_groups
select top 1000 * from tb_grouptypes
select top 1000 * from tb_groupentries

select * from tb_groups, tb_groupentries, tb_grouptypes
where tb_groupentries.groupid = tb_groups.id and tb_groupentries.entrytype = tb_grouptypes.dirid

--create a new group
insert into tb_groups (name) values ('testgroup')

--add 1 user to the new group (by id)
insert into tb_groupentries (groupid,entrytype,entryid) values
(
(select id from tb_groups where name = 'testgroup'),  --group id instered previously
1, --entry type. 1 means "endusers" (values from tb_grouptypes.dirid)
100207 --id from tb_users
)

--add all users to this group which username contains "111"
insert into tb_groupentries (groupid,entrytype,entryid)
select
(select id from tb_groups where name = 'testgroup'),
1,
id
from tb_users
where username like '%111%'

--select users from this group
select tb_users.id, tb_users.Username, tb_users.name
from tb_groupentries, tb_users
where
tb_groupentries.groupid = (select id from tb_groups where name = 'testgroup') and
tb_users.id = tb_groupentries.entryid

--delete group
delete from tb_groupentries where groupid = (select id from tb_groups where name = 'testgroup')
delete from tb_groups where name = 'testgroup'
```

5.98. Compatibility issues

In case if you have SIP signaling compatibility issues with old devices you might change these global config options:

addnewlineaftermsg = 2
rtpwritefirst = 0
allowheaders = 0

5.99. CLI settings / A and B numbers / Dial plan

There are multiple ways to apply your rules.

The server will "**normalize**" the numbers automatically based on rules from the global configuration (search for normalize). For example it can remove strange digits, IEC codes like 00 or +, enforce length, etc.
Also there are built-in rules for several countries, so users can dial without CC/IEC/NEC.

For outgoing calls if you need to use a tech prefix, you just have to enter the corresponding digits as tech prefix for the sip proxy user.

TechPrefix:
The server can authorize and/or route the traffic after the incoming techprefix.
Sip users can have techprefixes too. this is usually common for reseller company users.
If no techprefix is specified, then it will be loaded from tb_pxrules if any.
Sim owners and vpc users can have a list of prefixes separated by comma.
If no techprefix is specified, 111 will be inserted for incoming called numbers.
If the techprefix is „-1", then the original techprefix will be forwarded.
If the techprefix is „-2", then the original techprefix will be inserted in cdr record (but not forwarded).
If the techprefix is empty, then only the normalized callednumber will be forwarded.
The following techprefixes are reserved for the server: 111,222,999.
Only 3 digit techprefix is allowed. If your traffic sender needs another techprefix length, you must rewrite the incoming number in the "Prefix Rules" form.
Example: protocol: sip, Type: ip, value: your traffic sender ip, rewritefrom: oldtechprefix, rewriteto: newtechprefix.

If you need more **complex rules** then you can use the **prefix rules form** in the mmanage application. Here you can add/remove any prefix for A and B numbers before and after the routing.

You can rewrite prefixes before they arrive to the routing by entering your preferences here.
The Mizu routing engine will accept only 3 digit length techprefixes or no thechprefix, so you must convert them here if your traffic sender will send the traffic with techprefix that are not three digit length.

For example you can set up a rule which defines that every incoming number from ip 111.111.111.111 on H323 if begins with 1234 must be rewritten to begin with 56. Number 123499999 will be rewritten to 5699999.
If the "RewriteFrom" is emtly, then the "RewriteTo" fill be inserted before the number

To rewrite prefixes on router number normalization, you have to set the following global config values:
prefixrewritestr: the original prefix
prefixrewritefrom: keep from
prefixrewriteto: inserted string
for example to handle the hungarian roaming prefix: 08 + SK + BK + NSN +SN you have to set the following values:
prefixrewritestr: 08X…
prefixrewritefrom: 9
prefixrewriteto: 36

If you need more flexibility, you can edit the **v_check_pxrules** stored procedure manually using any SQL expression.

For **CLI** control you can set the “cli” field for any user:
CLI: CLIR and CLIP settings

0: forward always (forward asserted as normal number always!). Will not hide, even if caller was set so.
1: normal handling (forward asserted as normal number) -default
2: forward as asserted identity always (identityrewrite asserted)
3: forward as asserted identity only to trusted domains (identityrewrite asserted)
4: normal hide (no idenityrewrite forwarding)
5: force hide (no asserted identity too!). Always hidden.

To completely rewrite the **A number** you can use the “rewriteanumber” field.
To cut some prefixes from the A number use the “cutanumber” field.
To allow some A numbers and rewrite other numbers you can use the identityrewrite and identityforward fields.
This can be useful when not all your user have real PSTN numbers.

Addtechprefix: we insert this number before the callednumber if the caller doesn’t send its calls with tech prefix.
identityforward: we can toward these kinds of usernames and the other we rewrite to „identityrewrite”.
identityrewrite: if the caller username don’t match the identityforward prefix, then we rewrite it.

For more control over the A number you can use the prefix rules form or manually edit the v_check_pxrules stored procedure.

Prefixed can be also used for **authentication, routing and billing**.

5.100. How to rewrite caller/called numbers

- The server is configured by default to do some basic number normalization like removing junk characters from the numbers, removing + or 00 and a few others. To control this operation (change or turn on/off) you can change the normalizexxx **global configuration options**: normalize_clean, normalizenumbers, normalize_localpx, minnormalizelength, incomingprefix, outgoingprefix
- If you just need to add or remove a techprefix, then you should use the “**techprefix**” field associated to the caller/called (traffic sender / sip server). This is accessible on the Users and devices form -> Settings tab

- If more complex number rewrite rules are required, then you can use the Routing -> **Rules** form to define any conditional number rewrite.
- Other methods include simple prefix rewrite with “Prefix Rules” or from SQL with the predefined dialplan stored procedure accessible from the “Dial Plan” form (With this module you have direct access to the dialplan stored procedure and you can use the power of the SQL language to implement any kind of rules and number rewriting. For more details please read the dial plan section.)

It is a good practice to always have the IEC (international escape code: + ,00,011,etc) removed before the routing is done and add it back only for the outgoing routes when this is required (for example by filling the techprefix for that route). This will also simplify your routing and billing rules (you can have the price listing filled only with numbers that begins with the country code and not with IEC codes)

Please note that when you are using prefix based routing or billing, than usually the already modified (normalized) number are used by these modules. This means that for example if user will dial 1999999 and if you rewrite the 1 prefix to 44222 (resulting in 44222999999) then the routing and the billing will be done after 44222.. and not after the original number.

In the cdr record 3 called (B) number is stored:
-origcallernumber: this is the original called number arrived to our server
-callednumber: this is the normalized number (with applied number rewrite rules)
-dialednumber: this is the number sent out on the b call leg (this can have an additional techprefix for example)

For more details, see the following chapters:

- Caller ID Settings
- Rules

5.101. Reseller checklist

-set the “resellerbilling” global config option to true
-create at least one Owner account and put all top reseller below this account (needed for proper billing).
-unlimited reseller child/parent relationship can be created (limited by the “maxresellers” global config options)
-this relationship can be analyzed using the “Ownerships” form
-make sure that you have a public reseller price listing (enduser costs)
-reseller will be able to create their own prices on the website
-reseller can create a “base tariff” and other tariffs assigned to individual users
-individual reseller prices are stored in tb_billsources with their “resellerid”
-if reseller has not tariffs, than the billing will be done after usual enduserprices
-top reseller id is stored in tb_cdrs.resellerid and the “othercost” field will contain the payment from the reseller (loaded from public reseller price)
-individual reseller cdr records are stored in the tb_cdrresellers
-check reseller statistics on the “Statistics” form
-for earlypay set the reseller stage field to 9

5.102. Register timeout

-when a client send a register message, the server will try to enforce the timeout set by the “upperexpire“ global config option
-when the expire sent by the client is between “upperexpiremin” and “upperexpiremax” then it is accepted. Otherwise the “upperexpire” value is offered.
-the negotiated timeout is stored in the user record (“regival field”) and will be checked for timeouts and if the user is offline or online
-if no value is negotiated, than the client is considered offline after the period set by the “userofflinemin” global config option

The expire parameter in the contact field can be controlled by the sendbackregcontact global config option (0=no,1=without params like expire,2=append (modified) expire param,3=exactly)

For upper server registration, the mizu voip server will always try to negotiate the value set for the sip server user or if this is empty than by the REGISTERIVAL (or REGISTERIVAL*2 for servers) global config option, but it will accept also other values negotiated or enforced by the server side.

The maximum timeout for the lifetime of a registrar endpoint can be controlled with the "maxregelivetime" setting.
The user status is written to the database periodically driven by the expire interval. If the expire interval sent by the client is not valid, then the status is written in every “STATUSSAVEIVAL” minutes.

5.103. Anonymous users

One ore more of the following setting must be set to allow anonymous server acccess:
-set "open voip relay" for any enduser or traffic sender
-set the "freeivraccess" global config option
-set the "freeaccessuserid" global config option
- set the " allowanonymouscaller" global config option to "true"
- set the " allowanonymoususers " global config option to "1" or "2"
-an enduser named "sysanonymous" must be present
-set the " enforcestrongauth" global config option must be set to "false"
-set the "autocreatereguser" config option to 2 (for voip tunnels only)

5.100. How to play advertisement for the callers

This topic have been moved to the "How to play a voice file before calls" FAQ point.

5.101. Using TCP port 80/443

In a typical Mizu VoIP server a few application might have to use port 80 and/or 443. You might set up them to use different ports but from several countries or behind corporate firewalls only port 80 and 443 might be enabled for HTTP access. If more than one application have to use these ports, then you must bind them to different IP's.
The following applications might have to use port 80 and 443:
-The built-in HTTP interface. You can setup to listen on any port, but if PayPal is used, then it must be listening on port 80.
Use "httpserverbindip" and "httpserver" for the configuration.
-The web enduser portal. Can use any port.
Use "cfg_boundip" and "cfg_port" for the configuration.
-The HTTP tunnel. Can use any port but you should prefer ports 80 or 8080.
Use "httpserverbindip" and "httpserverbindip" for the configuration.
-The local IIS server
Use the following FAQ section to reconfigure its IP binding.
-The message queue service

Starting from v.6.2.2 (May 2015) the mizu VoIP server unified all services on port 80 and 443 (the universal main access port and secure port). This means that it is running multiple different services on the same port which is set to 80/443 by default and includes the followings:

- API: http://domain.com/mvapireq/?apientry=balance&authid=USRNAME
- console commands and the old http API (as a subset of the new API)
- paypal and other payment processing IPN: credit recharge answer
- enduser webportal: http://domain.com/webvoipportal
- built-in http server: http://domain.com/mvweb/anyfile.html (can server any file placed in the mvweb directory)
- internal websocket will route to API: http://domain.com/mvstwebsock
- websocket for webrtc: http://domain.com/mfstwebsock
- websocket for MMQ: http://domain.com/mmstwebsock
- tunneling/encryption over port 80 (VoIP over HTTP)

All the above services can be served also via SSL (https) which is running on port 443 by default. See the TLS FAQ for details.

5.102. How to bind IIS webservice to a single interface?

Window Server - Binding Web Interface to port 80 when IIS is running

By default IIS listens on all IP addresses on port 80, assigned to a PC. So what needs to be done is to configure IIS to listen only on one IP address. Follow the steps described below, depending on what operating system is running on the server:

Window Server 2003

Go to Administrative Tools->Internet Information Services, and then to "server name"->Web sites->Default Web site, right-click, properties, go to the 'Web Site' tab, and in 'Web Site Identification' put the IP address you want IIS to bind to or click the 'Advanced' button to select more than one IP.

Then disable 'Socket Pooling' by following the next steps:

1. Open a command prompt and make sure you are in the X:\Inetpub\Adminscripts folder (where X is the IIS installation drive).
2. After you open the Adminscripts folder, type the following line at the command prompt:

```
cscript adsutil.vbs set w3svc/disablesocketpooling true
```

The command replies: disablesocketpooling : (BOOLEAN) True

3. Stop and start the IIS Admin and depending services.

Window Server 2008/2012/2016/2019/2022/2025

Run this command from the command port:
netsh http add iplisten ipaddress=IPADDRESS (replace IPADDRESS with the IP to bind)

verify: netsh http show iplisten
delete old: netsh http delete iplisten ipaddress=IPADDRESS

http://msdn.microsoft.com/en-us/library/windows/desktop/cc307236%28v=vs.85%29.aspx

5.103. What are the typical bandwidth requirements for VoIP?

Voice payload (G.729): 20 bytes
RTP header: 12 bytes
UDP header: 8 bytes
IP header: 20 bytes
Total bandwidth = 480 bits / 20 ms

Total bandwidth = 24,000 bps

If 80% of the calls are using g729 and the rest g711:

1 channel bandwidth: 40 kbits = 13 GB / month
10 channel bandwidth: 400 kbits = 130 GB / month
100 channel: 4 mbits = 1300 GB / month (1.3 TB)
1000 channel: 40 mbits = 13000 GB / month (13 TB)
10000 channel: 400 mbits = 130000 GB / month (130 TB)
If both the traffic sender and the termination devices are located on remote peers, then you have to multiply these values with 2X (especially if you are billed for both “in” and “out” traffic)

Note:

1. These callculations are valid only if you have all the time X amount of traffic. This is usually not the case as you will have off-peak times.
2. These calculations are valid only if all your traffic needs RTP routing. Under normal circumstances you should be able to offload a lot of RTP routing from your server and enable the endpoints to communicate directly between them. This can be handled automatically by the Mizu VoIP server.

The bandwidth needed for g729 codec is around 32 kbits (this is the total bandwidth including RTP, UDP and Ethernet headers)
We assumed that not all calls will be handled by g729. Let's say around 20% is with g711. So let's calculate with 40 kbits in average.
This means 4 mbits/sec for 100 channels which is 0.5 MB/sec.
*We have 60*60*24 seconds in a day and approximately 60*60*24*30 seconds in a months which is 2592000 seconds.*
*So we have 0.5 * 2592000 MB/month which is 1296000 MB which is 1296 GB/month for if you have 100 simultaneous calls in average.*

5.104. How the softphone autoupgrade works?

The autoupgrade feature is enabled if the “AUTOUPDATEURL_DEF” configuration option is set to a valid HTTP URL.

Under this url 2 file have to be stored:
mizuupdate.zip which contains the files that have to be upgraded
mizuupdate.txt with the following content:

[autoupdate]
hasnewversion=true (set to false to disable)
newversion=x.x.x (new software version number. Clients will not upgrade if their version is equal or newer)
updatefor=* (old software version number that have to be upgraded or * for all)
mode=normal (normal or hidden)

comment=New version is available (any comment that will be presented for the user)

The mizuupdate.txt will be checked in the following circumstances:
If the AUTOUPDATEMODE is set to 1 (rare checks, this is the default setting):
-at least 1 minute have been elapsed since the softphone have been started or 1 day since the last check
-there is no calls in progress
-the softphone have been started at least 15X
-at least 20 days have been elapsed since the last software update if any
-at least 5 days have been elapsed since the last check
-there was at least 5 successful calls
-the users have not pressed the "later" button in the last 22 days
If the AUTOUPDATEMODE is set to 2 (more aggressive):
-at least 1 minute have been elapsed since the softphone have been started or 1 day since the last check
-there is no calls in progress
-the softphone have been started at least 3X
-at least 2 days have been elapsed since the last software update if any
-at least 1 days have been elapsed since the last check
-the users have not pressed the "later" button in the last 11 days

If a new version is found, than the softphone will open a new tab requesting user confirmation with a short description and Download/Later buttons.
If the user press the Download button, then the upgrade will be downloaded in the background. Once the softphone is restarted, the new version will be loaded.
If the user press the Later button, then the next check will be performed 11 or 22 days later depending on ther AUTOUPDATEMODE setting.

5.105. How to setup PayPal?

In short:
All you need to be able to accept PayPal payments are the followings:

1. Set your server address (IP or domain) on your PayPal account as the IPN URL.
2. Set your paypal ID as the "paypal_accountid" global config (search for "paypal" on the Configuration form).

Now your users can buy credit on from the web control panel or from inside the client applications (if this was integrated).
You can see the payment transactions on the "Reports" form.

Details:
You will need to open an account at https://www.paypal.com.
The **IPN url** must be set to your server HTTP interface which must run on port 80 for PayPal (usually your server IP or the IP address configured by the main port config option or via the httpserverbindip option if the unified main port is disabled). This configuration have to be done on your paypal account

under your “Profile” menu -> select “My selling tools” -> update “Instant payment notifications” and then set the “Notification URL” to point to your VoIP server HTTP API address (usually the IP address or domain name of your server) with mzpaypalpaymenthandler path.
For example: http://myserverip/mzpaypalpaymenthandler
You should also set the following URI’s:
Canceled URI: http://yourwebaddress.com?function=notify&flag=autologin&formparam=ppcancel&now=true
Successful URI: http://yourwebaddress.com?function=notify&flag=autologin&formparam=ppok&now=true
Paypal payments can be initiated from any web interface.
For the PayPal button customization please send your paypal “buy now” button html code to mizutech support. (the code must not be encrypted). Otherwise a default button will be used. You must also properly setup the “paypal” related global configuration options. At least the **paypal_emailid** and **the paypal_accountid** have to be set correctly (search for “paypal” on the “Configurations” form)
Make sure that your paypal account is “verified” before switching to production. For this you might have to send some documents to PayPal like your Driver ID. If the account is not verified, then PayPal might hold all transactions after a certain limit.
The amount applied for the user credit will be the real (net) amount received which is usually lower than the amount sent by the user due to paypal transaction fee and other additional costs (for example currency conversion). There is no known fix percent for this value but paypal will send it with the transaction and the vopip server will calculate accordingly.

Configuration details:
paypal_enabled: enable/disable the PayPal payment option. Enabled by default, but it is unusable until you set either the paypal_accountid or the paypal_emailid setting.
paypal_enabled_for_resellers: enable/disable separate PayPal account for resellers (enabled by default)
paypal_mustverify: request confirmation from paypal after each payment. Enabled by default
paypal_accountid: your paypal account id where the payments will be sent
paypal_emailid: your paypal email address where the payments will be sent
paypal_currency: the enforced currency. Otherwise the user currency setting is used
paypal_currencies: the list of accepted currencies. By default the followings are accepted: USD,EUR,GBP, your main currency
paypal_maxamount: maximum payment accepted. Default value is 600.
paypal_minamount: minimum payment accepted. Default value is 5
paypal_checkabove_verified: payments above this value must be verified first if the sender account is not “verified” by paypal (default value is 50)
paypal_checkabove_all: all payments above this value must be verified first (default value is 100)
paypal_feetype: which amount have to be added to the user credit: 0=really received (default),1=send by the user
paypal_addpercent: modify the received amount. Default value is 100 (no changes)
paypal_feepercent: add a fee percent (before payment to cover the paypal taxes, so the user will pay a bit more and this is visible to user)
paypal_feevalue: add a fee value (before payment to cover the paypal taxes, so the user will pay a bit more and this is visible to user)
paypal_sendemail: send emails about payments for the users
paypal_sendemailtoadmins: notify the admins about the new payments
paypal_sendemailtosupport: notify the support users about the new payments
paypal_emailonreceived: emai body text to be sent on received payments (a default message will be applied if this is not set)

paypal_emailonapprowed: emai body text to be sent on approved payments (a default message will be applied if this is not set)
paypal_verifysource: verify if the transaction was initiated from the webportal (0=no,1=if exists,2=always)
paypal_checkemail: check if the user email address is the same received from paypal. If not, then the payment is not applied automatically. Default is false.
paypal_firstismanual: first transaction have to be applied manually. Default is false.
paypal_checktrusted: auto allow transactions only for users on the trusted list. Otherwise the transaction have to be applied manually. Possible values: 0,1,2,3,4 (0=never,1=only if paypal_checkemail is true and no match or amount is high,2=always,3=always + email). Default is 1.
feature_paypalinnewwindow: will open the paypal payment in a separate page if set to true.

All PayPal transactions will be stored in the **tb_payprocessing** table which has the following fields:

- Userid: user database ID from tb_users
- Amount: paid ammount
- Currency: paypal currency
- Transactionid: unique id for all transactions
- Stage: 0=unknown,1=started,2=notification received,4=verified and applied,5=failed, 6=need to be accepted manually, 7=manually accepted, 8=manually rejected
- Comment: contains the reason of the failures and all parameters as received from paypal

Example query to list all transactions that need to be approved from the last week:

```
select top 1000 tb_users.Username, tb_users.credit as 'usercredit', tb_payprocessing.*
from tb_payprocessing with(nolock)
left join tb_users with(nolock) on (tb_users.id = tb_payprocessing.userid)
where
tb_payprocessing.stage = 6 and
tb_payprocessing.datum > GETDATE()-7
order by datum desc
```

The transactions can be viewed also from Mizu Manage -> Billing section -> Reports form (Transcations All/Success/Failed/etc) or right click on user form and select the “Payment history” from the popup menu.

Applied payments will be also stored in the **tb_invoices** table.

If you wish to automatically allow all transactions, set the paypal_checkemail and paypal_checktrusted settings to 0. Otherwise some transactions will have to be applied manually.

For a custom paypal payment start page (pay now button) you have to store it’s html as “paypaltemplate.txt” in the server directory.
Otherwise the default options will be presented.
You can specify the following keywords (replaced runtime): [ACCOUNTID], [USERID], [BACKBTN]

Common security templates:

Released -force the paypal payments to be accepted even from not trusted users, set the following settings:

-paypal_checktrusted: 0
-paypal_checkemail: false
-paypal_firstismanual: false
-might increase paypal_checkabove_verified
-might increase paypal_checkabove_all
-might increase paypal_maxamount
-might decrease paypal_minamount

Hardened –the first payment have to be approved manually:

-paypal_firstismanual: true

Hardened –auto allow only from users in the trusted list (tb_trustedusers):

-paypal_checktrusted: 2

Hardened –auto allow only from users in the trusted list and only if email address match (tb_trustedusers):

-paypal_checktrusted: 3
-paypal_checkemail: true

Note: PayPal payments can be initiated also from third party apps or web pages. Just use the "initpayment" API before to submit the payment to paypal and make sure that the IPN URI is set for automatic payment processing.

5.106. How to setup a registered SIP server

The Mizu server allows you to declare outbound trunk, which must register with SIP digest authentication to be able to send calls to.
This is useful if the outbound route is on a dynamic IP or behind NAT, thus the server can learn its address from the register session. It can be used also for non-reliable links that can go offline, such as remote SIP-GSM gateways.
This kind of outbound routes are to be configured as SIP Servers on the mizu server with the "Accept Register" option selected.
Step by step to-do:

1. Set the sipserverreg global config parameter after your needs from the "Configurations" form. You should set it to 2 if these routes also has some limited credit (such as SIM cards) or set to 1 otherwise. Possible values for sipserverreg: -1: def auto, 0: no, 1: allow register from SIP servers, 2: billing (decrease sipserver credit)
2. Add your trunk as a SIP server from the "Users and Devices" form. Enter the Username/Password which will be used for authentication.
3. Go to the Functions tab and select the "Accept Register" checkbox.
4. Add the SIP server to Routing.

In case if you need also credit calculation (for example to not allow more calls if credit is depleted):

5. Set the “sipserverreg” global config to 2.
6. On the “Users and devices” form set a credit value for the SIP server entry what you created earlier
7. Configure the billing as you wish from the “Price Setup” form (Type: Provider (called) Cost).

Old notes (deprecated):
If the outbound route is on a dynamic IP or behind NAT, then it must register to the server so the server can learn its address.
For this you must create a usual SIP server account and also an enduser account. The server will register to the user account which is “linked” with the SIP server account.
Use the “PRManager” field to link an enduser account to a sipserver (this have to be set for the enduser)
Set the “haslinkedaccounts” global config option to “true” for this to work.

5.107. Server clustering

You can setup multiple servers with one (clustered) database backend.
For this you will have to edit the .ini file near the server and set the node value. Each server must have its unique node id (1,2,3…)

```
[database]
node=1
```

For any global configuration option specific to a node, a #NODENUMBER have to be appended for the key.
For example bindip#1.
(Usually only the IP and ports are different for nodes and eventually the encv2 settings if encryptions are used. However, of course, you can set any settings do differ by node).

You can set one main node with the “mainnode” global config option which must be equal with a node id. If not set, then node 1 will be the main.
Certain tasks (maintenance, reporting, etc) are handled only by the mainnode. If the mainnode app server fails, then another mainnode must be assigned manually.

If a setting is not set for a node, then the node will load the default setting. However nodes will not save to the default space, only into their owns.

For database clustering please consult your db admin.

check:
  webportal, web interface
  check all db table, stored procedure, cpp file and server documentation
  check cdr and billing, asyncDecreaseCredit, check user credit
  select from tb_users order by instance

implementation:
db.h NODE, instanceclause, config.ismainnode
tb_users.srvinstance, tb_currcals.srvinstance, tb_cdrs.srvinstance
mizu clients will send \r\n for all servers to keep alive the nat, other calls have to be routed between server instances (tb_user.instance)
voicehere will work only on it's own server

5.108. Quality statistics (whithout extended cdrinfo)

```
--simple stats by call type
SELECT
b.marker, count(b.id) as 'CDRC (count)',
sum(b.realduration)/60 as 'SL (min)',
sum(SIGN(b.realduration))*100/count(b.id) as 'ASR (%)',
avg(case when b.realduration > 0 then b.realduration else null end) as 'ACD (sec)'
FROM tb_cdrs b WITH(NOLOCK) WHERE
b.datum >= :from AND b.datum <= :to
GROUP BY  b.marker   ORDER BY b.marker


--stats by call quality
SELECT
b.marker, count(b.id) as 'CDRC (count)',
sum(b.realduration)/60 as 'SL (min)',
sum(SIGN(b.realduration))*100/count(b.id) as 'ASR (%)',
avg(case when b.realduration > 0 then b.realduration else null end) as 'ACD (sec)' ,
avg(
      case
      when realduration > 5 and realduration < 35 and (rtpsent > 400 or rtprec > 400) and
      (
        (rtpsent >= 0 and rtpsent < rtprec/9) or
        (rtprec >= 0 and rtprec < rtpsent/9) or
        (rtplost > rtpsent/2)
      )
      then 0
      when realduration > 5 and realduration < 55 and (rtpsent > 400 or rtprec > 400) and
      (
        (rtpsent >= 0 and rtpsent < rtprec/2) or
        (rtprec >= 0 and rtprec < rtpsent/2) or
        (rtplost > rtpsent/4)
      )
      then 2
      when realduration > 55 and rtpsent > 2000 and rtprec > 2000 and
      (
        (rtpsent > rtprec/2) or
        (rtprec > rtpsent/2) or
        (rtplost < rtpsent/6)
      )
      then 8
      when realduration > 120 and rtpsent > 2000 and rtprec > 2000 and
      (
        (rtpsent > rtprec/2) or
        (rtprec > rtpsent/2) or
        (rtplost < rtpsent/6)
      )
      then 10
      else null end
) as 'QUALITY (0 = worst,10=best)'
FROM tb_cdrs b WITH(NOLOCK) WHERE
b.datum >= GETDATE()-1
and b.realduration > 0
GROUP BY  b.marker   ORDER BY b.marker

--cdr list
select top 10 marker,connecttime,realduration,rtpsent,rtprec,rtplost,
case
when realduration > 5 and realduration < 35 and (rtpsent > 400 or rtprec > 400) and
(
  (rtpsent >= 0 and rtpsent < rtprec/9) or
  (rtprec >= 0 and rtprec < rtpsent/9) or
  (rtplost > rtpsent/2)
)
then 'NOVOICE'
when realduration > 5 and realduration < 55 and (rtpsent > 400 or rtprec > 400) and
(
  (rtpsent >= 0 and rtpsent < rtprec/2) or
  (rtprec >= 0 and rtprec < rtpsent/2) or
  (rtplost > rtpsent/4)
)
then 'LOWQUALITY'
when realduration > 40 and rtpsent > 2000 and rtprec > 2000 and
(
  (rtpsent > rtprec/2) and
  (rtprec > rtpsent/2) and
  (rtplost < rtpsent/6)
)
then 'GOODQUALITY'
when realduration > 140
then 'GOODQUALITY'

else 'UNKNOWN' end
from tb_cdrs with(nolock)
where datum > getdate()-1
and realduration > 3
order by datum desc
```

5.109. How to obtain geolocation and advanced call quality statistics

1. Import GeoInfo database with MManage -> Config menu -> Database -> Setup -> Setup GeoIP database (GeoIP.exe)
   At least the countries must be imported
2. If you need geoinfo data from previous days, then run the "createcdrinfo" command from the server console with a "number of days" parameter. This will take 5-10 minutes (don't run on high server load)
3. Make sure that the database has a table called tb_cdrinfo and the "cdrinfo" global config option is set to 1,2 or 3.
4. Once this tables and configurations are set, the server will populate the tb_cdrinfo table automatically and after some time you can run queries like below:

Quality statistics:

```
SELECT
b.marker,
count(b.id) as 'CDRC (count)',
sum(b.realduration)/60 as 'SL (min)',
sum(SIGN(b.realduration))*100/count(b.id) as 'ASR (%)',
avg(case when b.realduration > 0 then b.realduration else null end) as 'ACD (sec)' ,
AVG((i.srvlosspercent + i.clientlosspercent + i.totallosspercent)/3) as 'packetloss (%)',
avg(
 case
when realduration > 5 and realduration < 35 and (i.srvsent > 400 or i.srvrec > 400) and
(
  (i.srvsent >= 0 and i.srvsent < i.srvrec/10) or
  (i.srvrec >= 0 and i.srvrec < i.srvsent/10) or
  i.srvlosspercent > 50 or i.clientlosspercent > 50 or i.totallosspercent > 50 or
  (i.srvlost > i.srvsent/2)
)
then 0
when realduration > 5 and realduration < 55 and (i.srvsent > 400 or i.srvrec > 400) and
(
  (i.srvsent >= 0 and i.srvsent < i.srvrec/4) or
  (i.srvrec >= 0 and i.srvrec < i.srvsent/4) or
  i.srvlosspercent > 20 or i.clientlosspercent > 20 or i.totallosspercent > 20 or
  (i.srvlost > i.srvsent/4)
)
then 2
when realduration > 40 and i.srvsent > 2000 and i.srvrec > 2000 and
(
  (i.srvsent > i.srvrec/4) and
  (i.srvrec > i.srvsent/4) and
  i.srvlosspercent < 13 and i.clientlosspercent < 13 and i.totallosspercent < 13 and
  (i.srvlost < i.srvsent/6)
)
then 8
when realduration > 65 and i.srvsent > 2000 and i.srvrec > 2000 and
(
  (i.srvsent > i.srvrec/4) and
  (i.srvrec > i.srvsent/4) and
  i.srvlosspercent < 5 and i.clientlosspercent < 5 and i.totallosspercent < 5 and
  (i.srvlost < i.srvsent/6)
)
then 10
when realduration > 180 and i.srvlosspercent < 15 and i.clientlosspercent < 15 and i.totallosspercent < 19
then 10
else null end
) as 'QUALITY (0 = worst,10=best)'
FROM tb_cdrs b WITH(NOLOCK)
left join tb_cdrinfo i with(nolock)  on (b.id = i.cdrid )
left join tb_users u1 with(nolock)  on (b.callerid = u1.id )
left join tb_users u2 with(nolock) on (b.calledid = u2.id )
left join tb_reasoncodes r with(nolock) on (b.discreason = r.code)
WHERE
b.datum >= GETDATE()-1
and b.realduration > 0
GROUP BY  b.marker   ORDER BY b.marker
```

CDR List with details geolocation and quality statistics:

```
SELECT top 100
 b.id,   b.datum, b.connecttime as 'Connect', b.realduration as 'duration',     u1.username as 'caller', i.geoinfo
as 'caller location',   u2.username as 'called',    b.callednumber as 'callednumber',
 case when (select top 1 tb_users.type from tb_users with(nolock) where b.calledid = tb_users.id) = 0 then 'LOCAL'
else 'OUT' end as 'DIR',
 (select top 1 tb_directions.name from tb_directions with(nolock) where b.dirid = tb_directions.id) as 'called
location',
 r.text as 'discreason' , (i.srvlosspercent + i.clientlosspercent + i.totallosspercent)/3 as 'packetloss %',
 case
when realduration > 5 and realduration < 35 and (i.srvsent > 400 or i.srvrec > 400) and
(
  (i.srvsent >= 0 and i.srvsent < i.srvrec/10) or
  (i.srvrec >= 0 and i.srvrec < i.srvsent/10) or
  i.srvlosspercent > 50 or i.clientlosspercent > 50 or i.totallosspercent > 50 or
  (i.srvlost > i.srvsent/2)
)
then 'NOVOICE'
when realduration > 5 and realduration < 55 and (i.srvsent > 400 or i.srvrec > 400) and
(
  (i.srvsent >= 0 and i.srvsent < i.srvrec/4) or
  (i.srvrec >= 0 and i.srvrec < i.srvsent/4) or
  i.srvlosspercent > 20 or i.clientlosspercent > 20 or i.totallosspercent > 20 or
  (i.srvlost > i.srvsent/4)
)
then 'LOW'
when realduration > 40 and i.srvsent > 2000 and i.srvrec > 2000 and
(
  (i.srvsent > i.srvrec/4) and
  (i.srvrec > i.srvsent/4) and
  i.srvlosspercent < 10 and i.clientlosspercent < 10 and i.totallosspercent < 10 and
  (i.srvlost < i.srvsent/6)
)
then 'GOOD'
when realduration > 140 and i.srvlosspercent < 15 and i.clientlosspercent < 15 and i.totallosspercent < 19
then 'GOOD'
else 'UNKNOWN' end as 'QUALITY',
b.marker
FROM tb_cdrs b WITH(NOLOCK)
left join tb_cdrinfo i with(nolock)  on (b.id = i.cdrid )
left join tb_users u1 with(nolock)  on (b.callerid = u1.id )
left join tb_users u2 with(nolock) on (b.calledid = u2.id )
left join tb_reasoncodes r with(nolock) on (b.discreason = r.code)
WHERE
b.datum >= GETDATE()-1
and b.realduration > 0
ORDER by b.datum desc
```

5.110. Database file usage

```
SELECT CAST(DB_NAME(mstr.database_id) AS VARCHAR(24)) AS 'Database',
       CAST(mstr.physical_name AS VARCHAR(72)) AS 'File',
       stats.num_of_reads,
       stats.num_of_bytes_read/1000000 as 'MB read',
       stats.num_of_writes,
       stats.num_of_bytes_written/1000000 as 'MB write',
       stats.size_on_disk_bytes/1000000 as 'MB size'
FROM sys.dm_io_virtual_file_stats(null,null) AS stats
         join sys.master_files AS mstr
              ON mstr.database_id = stats.database_id
                 and mstr.FILE_ID = stats.FILE_ID
WHERE DB_NAME(mstr.database_id) = 'tempdb' or
      DB_NAME(mstr.database_id) = 'mserver';
go
```

5.111. How to recalculate the prices for certain cdr

This can be done with the “rebill” command from the server console.
From the MizuManage open the Server Console form and click on connect.
Then enter rebill,days,datefrom,dateto,usertype,childfor

For example the following command will recalculate all pricing for endusers whose parent is user with id 854 between Sept 1 and 30 in 2011.
rebill,,9/1/2011 00:00:00,9/31/2011 23:59:59,0,854

The following command will recalculate the billing for all users in the last 2 days (48 hours):

```
rebill,2
```

5.112. How to create different webportal for different resellers

Under the server app directory create a new folder and inside this folder create separate folders for each of your resellers which needs a separate webportal.
Copy the following files in these folders:
-webservice.exe
-mizuserver.ini
-the “Files” directory
-loginimage.jpg or loginimage.gif
- webphone.jar

In the mizuserver.ini leave the [database] settings as the main mizuserver.ini and change the [webportal] settings as follows:

```
mainportalreseller=RESELLER USER ID
cfg_port=WEB PORT
cfg_boundip=WEB IP
cfg_appname=app name
base_design=1
feature_loginpicture=true
loginpage_title=brandname
loginpage_footer1=any branding text
loginpage_footer2= any branding text
color_main_background=13421772
startpagelinkname=any link name
startpagelinkuri=any link uri
startpagelinkname2= any link name
startpagelinkuri2= any link uri
```

5.113. Database SQL fine-tune

Create a tb_tracing table with SQL Server Profiler, then run the followings:

create column TextData2 varchar(max)

```
update tb_tracing set TextData2 = TextData
```

(unselect the “Prevent saving changes that require table recreation” checkbox from Tools -> Options -> Designers)

add index for cpu and duration

cpu, duration,TextData2

```
select top 200 * from tb_tracing
order by duration desc
```

```
select top 200 * from tb_tracing
order by cpu desc
```

```
select sum(cpu), applicationname
from tb_tracing
group by applicationname
order by sum(cpu) desc
```

```
select sum(duration), applicationname
from tb_tracing
group by applicationname
order by sum(duration) desc
```

```
select sum(cpu), applicationname, substring(textdata2,1,29)
from tb_tracing
group by  applicationname, substring(textdata2,1,29)
order by sum(cpu) desc
```

```
select sum(duration), applicationname, substring(textdata2,1,29)
from tb_tracing
group by  applicationname, substring(textdata2,1,29)
order by sum(duration) desc
```

```
select sum(cpu), substring(textdata2,1,29)
from tb_tracing
group by  substring(textdata2,1,29)
order by sum(cpu) desc
```

5.114. Too slow UDP troughput

Modify your ethernet card settings:

Interrupt Moderation -->disable
flow control--->disable
checksum offload--->disable

5.115. Pricing speedup

If you have (or will plan to add) many prefix entries for multiple billing packets than you should consider to change to the int based algorithm especially if you are using LCR.

Follow these steps:

1. Create tb_billingtimes_int table with the prefix field set to int type if not already exists
2. Check the “Limitations” listed below
3. Migrate all entries to the new table:

```
INSERT INTO [msrv_2].[dbo].[tb_billingtimes_int]
           ([entryid]
           ,[isdiff]
           ,[timetype]
           ,[fromday]
           ,[fromhour]
           ,[frommin]
           ,[today]
           ,[tohour]
           ,[tomin]
           ,[prefix]
           ,[origprice]
           ,[price]
           ,[dirid]
           ,[billingstep2]
           ,[minammount2])
```

```
select
                                        [entryid]
     ,[isdiff]
     ,[timetype]
     ,[fromday]
     ,[fromhour]
     ,[frommin]
     ,[today]
     ,[tohour]
     ,[tomin]
     ,CAST ( prefix AS int )
     ,[origprice]
     ,[price]
     ,[dirid]
     ,[billingstep2]
     ,[minammount2]
from tb_billingtimes_int
```

If the query fails, check the below Limitations (or use TRY_CAST of best effort)

4. Change the function call v_getpriceexf to v_getpriceexf_int in the v_selpattern2 stored procedure (or disable LCR for further speedup)
5. Set the "usevgetpriceint" global config option to true (also requires dbversion 124) and reload or restat the sever
6. Restart MManage

Limitations:

- prefixes can be only numbers (this should be the case anyway)
- prefixes can't begin with 0, 00 or + (this should be avoided anyway)
- prefixes with more then 9 digits are not supported (max int is 999999999)
- for wildcard use -1 instead of *

Note: if you have lots of prefixes with more then 9 digits which can't be shortened, then it is possible to use bigint instead of int (which can hold 19 digits)

In this mode the _int version of the stored procedures will be used (v_getprice_int and fgetprice_int)

You can verify entries which doesn't match this criteria with the following query:

```
select top 1000 tb_billentries.name, tb_billingtimes.prefix
from  tb_billingtimes with(nolock),tb_billentries with(nolock)
where
tb_billingtimes.entryid = tb_billentries.id and
(
       prefix = '*' OR
       LEN(prefix) > 9 OR
       prefix like '0%'  OR
       ISNUMERIC(prefix) <> 1 OR
       TRY_CAST(prefix as INT) is null
)
order by tb_billentries.name, tb_billingtimes.prefix
```

Then you can resolve the issues in the following ways:

- for wildcard use -1 instead of * : `update tb_billingtimes set prefix = '-1' where prefix = '*'`
- prefixes can’t begin with 0, 00 or +: `update tb_billingtimes set prefix = SUBSTRING(prefix, 2, 99) where prefix = '0'` (run this multiple times as needed)
- prefixes with more then 9 digits are not supported: `update tb_billingtimes set prefix = SUBSTRING(prefix, 1, 9) where LEN(prefix) > 9`
  If you get duplicates error, then you might delete duplicates first with this query:

  ```
  delete from tb_billingtimes
  where LEN(prefix) >= 9 and
  id not in
  (
          select distinct min(id)
          from tb_billingtimes
          where LEN(prefix) > 9
          group by entryid, SUBSTRING(tb_billingtimes.prefix, 1, 9)
  )
  ```
- prefixes can be only numbers: check such number and fix manually or with a query after your needs. Or you might delete the with the following query: `delete from tb_billingtimes where ISNUMERIC(prefix) <> 1 OR TRY_CAST(prefix as INT) is null`

5.116. Shared DID’s

With the shared DID feature the same DID number can be assigned to different users (from webportal -> phonebook page). These will ease the access to users favorites from third-party networks (calling via third-party servers or via the IVR).

You can provide shared DID numbers for your users in the following way:
-Add a few DID number from the “Users and devices” form. These are like normal endusers but their username is a DID number and the “Shared DID” option is checked on the “Functions” tab (this way the “ispublic” field will be set to 4 for these numbers/users)
-Set the “feature_shareddidnumbers” configuration option to 1 (the default is -1 which means that will be automatically enabled once you have added a few DID numbers, otherwise automatically disabled)
-Once this is done, users can assign DID numbers for their Phonebook entries from the web enduser interface. (stored in the “udid” field in the “tb_speeddial” table)

5.117. Server supervisor

The mserver process has a builtin supervisor wich will do some integrity and whelth tests periodically and will do the required action on test failure.
In addition to this builtin procedure, you can start the vsupervisor service.
The following settings are defined in the "supervisor" section:

canrestartformalfunctions: if supervisor can do restarts (othervise only logs or emails the error). set to 0 to disable restarts default is 1
checklogs: if to check logs. if set to false, then "maxnologival" will have no effect. Default is true.
checkcpu: if to check cpu ussage
checkcallerids: check only cdr records that belongs to this caller. Example: 123,124
checkprefixes: check only cdr records only with this techprefix. Example: '333','555'

maxnocdrmin: restart if no cdr records for this number of minutes. Default is 20 (minutes).
minacd or minacl: min ACD treshold. Default is 15 sec.
minasr: min ASR treshold. Default is 3%.
maxnologival: maximum number of minutes without new record in tb_logs. Default is 15 min.
minactivesims: minumum number of active simcards. Default value is 0.
minactiveendusers: min active endusers (with statusdate set in the last 20 min). Default value is 1.
minactiveusers: min active users (some activity in the last 2 hour). Default value is 1.

peaktimebegin or peaktimebegintr: peaktime start hour. Default is 9.
peaktimeend or peaktimeendtr: peaktime end hour. Default is 19.
weekendispeak: treat weekend as peaktime (same traffic ammount). Default is false.
restartatnight: if to restart at every night. Default is false.
restartpcatfirst: don't restrart the service. Restart the pc immediately.
trafficammount: 0=after setup

all values will be altered in offpeak and weekend times after these rules:

```
    maxnocdrmin2*=60;
    maxnologival2*=3;
    minasr2 = minasr2/2;
    minacl2 = minacl*2/3;
    minactivesims/3;
```

*offpeak and weekend times are calculated based on peaktimebegin, peaktimeend, weekendispeak

The following programs are controlled (configurable):

```
        mserver_name = "mserver";
        atarongk_name = "atarongk";
        fvoipgsmgw_name = "fvoipgsmgw";
        vsip_name  = "vsip";
```

```
          valert_name = "valert";
  MSSQLSERVER_name = "MSSQLSERVER";
```

Alerting can be configured as described here.

5.118. How to check the current load on MS-SQL (query list)

```
select
    p.spid
,   right(convert(varchar,
            dateadd(ms, datediff(ms, P.last_batch, getdate()), '1900-01-01'),
            121), 12) as 'batch_duration'
,   P.program_name
,   P.hostname
,   P.loginame
from master.dbo.sysprocesses P
where P.spid > 50
and      P.status not in ('background', 'sleeping')
and      P.cmd not in ('AWAITING COMMAND'
                    ,'MIRROR HANDLER'
                    ,'LAZY WRITER'
                    ,'CHECKPOINT SLEEP'
                    ,'RA MANAGER')
order by batch_duration desc
```

```
declare
    @spid int
,   @stmt_start int
,   @stmt_end int
,   @sql_handle binary(20)

set @spid = 124 -- Fill this in

select  top 1
    @sql_handle = sql_handle
,   @stmt_start = case stmt_start when 0 then 0 else stmt_start / 2 end
,   @stmt_end = case stmt_end when -1 then -1 else stmt_end / 2 end
from    master.dbo.sysprocesses
where   spid = @spid
order by ecid
```

```
SELECT
    SUBSTRING(  text,
                COALESCE(NULLIF(@stmt_start, 0), 1),
                CASE @stmt_end
                        WHEN -1
                                THEN DATALENGTH(text)
                        ELSE
                                (@stmt_end - @stmt_start)
                        END
        )
FROM ::fn_get_sql(@sql_handle)
```

5.119. Load data from all tables from the database (for multiple servers)

The following query list all bind IP:

```
CREATE TABLE #tmp_333 (DatabaseName VARCHAR(50), valstr VARCHAR(64),keystr VARCHAR(50),inisection VARCHAR(50));

DECLARE DBCursor CURSOR
FOR
    SELECT  Name
    FROM    MASTER.dbo.sysdatabases
        WHERE name NOT IN ('master','model','msdb','tempdb');
OPEN DBCursor;

DECLARE @DBName VARCHAR(200) = '';
FETCH NEXT FROM DBCursor INTO @DBName;

WHILE @@FETCH_STATUS = 0
    BEGIN
        DECLARE @SQL NVARCHAR(MAX) = N'USE ' + QUOTENAME(@DBName) + '
                                        IF EXISTS
                                            (   SELECT  1
                                                FROM    sys.tables
                                                WHERE   [Object_ID] = OBJECT_ID(N''dbo.tb_settings'')

                                            )
                                            BEGIN
                                                INSERT #tmp_333 (DatabaseName, valstr, keystr, inisection)
                                                SELECT @DB, [valstr], [keystr], [inisection]
                                                                                                  FROM    dbo.tb_settings
                                                                                                          WHERE keystr = ''bindip'' AND
inisection=''settings''
                                            END';
        EXECUTE SP_EXECUTESQL @SQL, N'@DB VARCHAR(200)', @DBName;
        FETCH NEXT FROM DBCursor INTO @DBName;
    END

CLOSE DBCursor;
DEALLOCATE DBCursor;

SELECT  *
FROM    #tmp_333;

DROP TABLE #tmp_333;
```

5.120. Number of services limit on a server

You might have to change the windows desktop heap settings if you are running multiple service instances (more than 10) on the same server.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\SubSystems\Windows

More details:
https://stackoverflow.com/questions/507853/system-error-code-8-not-enough-storage-is-available-to-process-this-command

http://ss64.com/nt/syntax-desktopheap.html
http://social.technet.microsoft.com/Forums/en-GB/winservergen/thread/c53e3c42-a798-417a-9c48-cd2b442310f2

http://blogs.msdn.com/b/ntdebugging/archive/2007/01/04/desktop-heap-overview.aspx
http://blogs.msdn.com/b/ntdebugging/archive/2007/07/05/desktop-heap-part-2.aspx

5.121. How to rewrite SMS number prefix

Create a stored procedure named v_dialplansms with the following inputs:

- tonum varchar(256)
- fromid int
- fromnum varchar(64)
- clientip varchar(64)

This sp must return one field containing the rewritten target number (“tonum”)

Set the “usesmsprefixsp” global config option to “true” and reload.

See the SMS chapter for more details.

5.122. How to limit the maximum monthly usage for postpaid accounts

Moved to “Security and account limiting” chapter.

5.123. Not enough storage is available to process this command

If you see similar errors, try to increase the IRPStackSize in the registry as described here:
http://blog.rongabriel.com/2010/03/08/not-enough-storage-is-available-to-process-this-command/

5.124. Enable session-timer

With session timers (RFC 4028) you can prevent orphaned sessions (hung calls).
Session timers are not enforced by default to prevent any complications or incompatibilities on a new server.
If all the media are routed trough your server, then there is no much reason to enable session timers because the media timeout can usually prevent orphaned calls. However if your server doesn’t need to route the media and you see orpaned calls (long calls in the current calls form which are no longer valid) then you should enabled this feature.
To avoid incompatibilities with all kind of client devices not supporting session timers or reinvite, you should enable session timers only for SIP server and Traffic sender users (or just for the devices generating orphaned calls). Just set the “sessiontimer” field to a value higher than 0:

- -2: never-ever use, even when pushed by the client
- -1: never use
- 0: don’t use (Default)
- 1: load from global config
- 2: yes if the peer has timer support
- 3: always yes, forced
- other: use with the specified timeout (minutes). Not recommended

Also please review the related global configuration settings (search for “sessiontimer” on the Configurations form):

- sessiontimer: 0: don’t use,1: load from global config ,2: autodetect (and using the sesskeepalive interval from the global configuration),3: autodetect with turned on by default, Other: use with the specified timeout (minutes)
- sessiontimerhandlelocally: wheter the server should handle or just bypass. 0=no,1=auto,2=forward,3=yes
- def_min_sessiontimer: minimum allowed value in seconds (sent also as “Min-SE” with the sip signaling). Must be higher then 3.
- def_mid_sessiontimer: recommended value in seconds
- def_max_sessiontimer: maximum accepted value in seconds

Recommended configuration:

- If there are no orphaned calls or media is routed leave the configuration with the default values (with the sessiontimer set to 0 which means disabled)
- If there are some orphaned calls set the “sessiontimer” field to 3 for SIP servers or Traffic senders generating orphaned calls.
- If there are a lot of orphaned calls (it can happen if the server network connection is not stable) set the “sessiontimer” field to 3 in the global settings (Configurations form)

5.125. Reseller rights

You can use the tb_users rigths field to limit the reseller rights (affecting mostly the webportal user interface):
0: minimal rights
8: can’t edit tariffs and can’t add resellers
10: can’t edit tariffs
12: can’t add resellers
30: normal rights (default)
40: increased rights
60: full rights

5.126. Server ini file

The database connection settings are stored in a configuration file which should be named “mizuserver.ini” and located near the service executable (mserver.exe). This file is readed when the server is starting so it can find its database (all other data and settings are stored in database).
The format is the following:

```
[database]
location=DB SERVER IP
port=DB SERVER PORT (USUALLY 1433 BUT OFTEN REWRITTEN TO 2223)
name=DATABSE NAME
username=USERNAME
password=PASSWORD
sqlinstancename=ENTER INSTANCE NAME IF YOU NEED PIPE ACCESS (. for the default mssql instance or np:\\.\pipe\sql\query)
conntype=SPECIFY HOW TO CONNECT (0=tcp only,1=pipe only,2=auto tcp first,3=auto pipe first if possible (default), 4=auto pipe first)
```

Example:

```
[database]
location=127.0.0.1
port=2223
name=mserver
username=sa
password=sqlsapassword
```

```
sqlinstancename=TESTSERVER\SQLEXPRESS
```

http://technet.microsoft.com/en-us/library/ms130822.aspx

5.127. Prices for toll free numbers

To increase the prices for toll free numbers you have to perform the following steps:

1. Create a new price list.
2. Assign the newly created price list to the end user (8557597386).
   You can do this in the "Users and Devices" section under the "Billing" tab.

Please note, that I have already created a new billing packet and assigned it to the number.
You just have to upload the prices.

To charge only connected calls, you have to apply the billing packet for B-leg only.
You can set this option on the "Price Setup" form in the "Applied for" section.

5.128. Server API

Please find the details in the VoIP_Server_API.pdf

5.129. External service, database or API

You can use the following global config options to setup external sources for user management, authentication, routing and billing:

```
int extern_cdr;
int extern_billing;
int extern_auth; //0=no,1=insert only,2=yes,3=check only external
int extern_routing;
String extern_connection;
String extern_connection_ivr;
String extern_connection_cdr;
String extern_connection_billing;
String extern_connection_auth;
String extern_connection_routing;
String extern_connection_usersync;
String extern_sql_cdr;
String extern_sql_billing;
```

```
 String extern_sql_auth;
 String extern_sql_routing;
 String extern_sql_usersync_all;
 String extern_sql_usersync_new;
 String extern_http_address;
 String extern_http_cdr_address;
 String extern_http_billing_address;
 String extern_http_auth_address;
 String extern_http_routing_address;
 String extern_http_cdr;
 String extern_http_billing;
 String extern_http_auth;
 String extern_http_routing;
 String extern_http_usersync_all;
 String extern_http_usersync_all_address;
 String extern_http_usersync_new;
 String extern_http_usersync_new_address;
 bool extern_usersync;
```

5.130. How to setup Moneybookers (Skrill)

You will need to open a merchant (For Business) account at https://www.moneybookers.com.
Moneybooker payments can be initiated from any web interface.
At least the moneybookers_emailid, moneybookers_accountid and the moneybookers_notifyurl have to be set correctly (search for “moneybookers” on the “Configurations” form).

Configuration details:
moneybookers_enabled: enable/disable the Moneybookers payment option. Enabled by default, but it is unusable until you set the moneybookres_emailid and moneybookers_notifyurl setting.
moneybookers_enabled_for_resellers: enable/disable separate MoneyBookers account for resellers (enabled by default)

moneybookers_emailid: your moneybookers email address where the payments will be sent
moneybookers_notifyurl: the IPN (Instant Payment Notification) url, must be the same as the webportal address: http://WEBPORTAL_ADDRESS:PORT. Port must be specified only if other than 80.
moneybookers_accountid: this is your Moneybookers "Customer ID".
moneybookers_verifysource: verify if the transaction was initiated from the webportal (0=no,1=if exists,2=always)
moneybookers_mustverify: check every payment if it came from moneybookers. For this to work, a "secret word" has to be set in your Moneybookers account and in moneybookers_secretword setting. Enabled by default
moneybookers_secretword: set it in your Moneybookers account -> 'Merchant Tools' and in server (“Configurations” form)

moneybookers_checkemail: check if the user email address is the same received from moneybookers. If not, then the payment is not applied automatically (deprecated by checktrusted)
moneybookers_checktrusted: auto allow transactions only for users on the trusted list. Otherwise the transaction have to be applied manually. Possible values: 0,1,2,3,4 (0=never,1=only if no email match,2=always,3=always + email)
moneybookers_currency: the enforced currency. Otherwise the user currency setting is used
moneybookers_currencies: the list of accepted currencies. By default the followings are accepted: USD,EUR,GBP, your main currency
moneybookers_maxamount: maximum payment accepted. Default value is 600.
moneybookers_minamount: minimum payment accepted. Default value is 5
moneybookers_checkabove_notverified: payments above this value must be verified first if the sender account is not “verified” by moneybookers (default value is 50)
moneybookers_checkabove_all: all payments above this value must be verified first (default value is 100)
moneybookers_addpercent: modify the received amount. Default value is 100 (no changes)
moneybookers_feepercent: add a fee percent (before payment)
moneybookers_feevalue: add a fee value (before payment)
moneybookers_sendemail: send emails about payments for the users
moneybookers_sendemailtoadmins: notify the admins about the new payments
moneybookers_sendemailtosupport: notify the support users about the new payments
moneybookers_emailonreceived: emai body text to be sent on received payments (a default message will be applied if this is not set)
moneybookers_emailonapprowed: emai body text to be sent on approved payments (a default message will be applied if this is not set)

All Moneybookers transactions will be stored in the tb_payprocessing table which has the following fields:
Userid: user database ID from tb_users
Amount: paid ammount
Currency: user currency
Transactionid: unique id for all transactions
Stage: 0=unknown,1=started,2=notification received,4=verified and applied,5=failed, 6=need to be checked (no email match)
Comment: contains the reason of the failures and all patameters as received from Moneybookers

Example query to list all transactions that need to be approved from the last week:

```
select top 1000 tb_users.Username, tb_users.credit as 'usercredit', tb_payprocessing.*
from tb_payprocessing with(nolock)
left join tb_users with(nolock) on (tb_users.id = tb_payprocessing.userid)
where
tb_payprocessing.stage = 6 and
tb_payprocessing.datum > GETDATE()-7
order by datum desc
```

The transactions can be viewed also from Mizu Manage -> Billing section -> Reports form (Transcations All/Success/Failed/etc)

Applied payments will be also stored in the tb_invoices table.

If you wish to automatically allow all transactions, set the moneybookers_checkemail and moneybookers_checktrusted settings to 0. Otherwise some transactions will have to be applied manually.

5.131. Reliability –strict locking

The Mizu VoIP server by default uses opportunistic locking and lock-free data structures. Change the following settings for stricter locking:
usesockrecmutex = true
usertpmutex = true

Note: these setting might have a direct impact on performance

5.132. How to verify audio quality

The Mizu VoIP server by default uses opportunistic locking and lock-free data structures. Change the following settings for stricter locking:

1. Set the server to route the RTP (RTP Route caller/called: Always)
2. Start wireshark on the server
3. Make a call from softphone
4. Find the call-id (from softphone or sever logs).
5. Apply the following filter for wireshark: ((ip.src==SERVERIP || ip.dst==SERVERIP) && (ip.src==SOFTPHONESIPIP || ip.dst==SOFTPHONESIPIP || ip.src == SOFTPHONESDPIP || ip.dst==SOFTPHONESDPIP)) && (sip || rtp || rtcp) && (sip.Call-ID == "CALLID" || (udp.port == SOFTPHONERTPPORT || udp.port == SERVERRTPPORT))
   Replace the followings:
   - SERVERIP: the IP address of the VoIP server
   - SOFTPHONESIPIP: the IP address from where the INVITE was received from
   - SOFTPHONESDPIP: the IP address in the INVITE SDP (connect address c=)
   - CALLID: the SIP call-id
   - SOFTPHONERTPPORT: the RTP port from the INVITE SDP (media= line)
   - SERVERRTPPORT: the RTP port from the 200 OK SDP (media= line)

   (To easily find the SDP address you might first just filter to sip.Call-ID == "CALLID")
6. Wireshark -> Telephony menu -> RTP -> Show all streams
7. Check the statistics if there are packet loss or similar issues
8. Click on Analyze, then Save Paypload (save as raw.EXT)
   The following extensions (EXT) can be used:
   g9: for G.729 codec
   g3: for G.723 codec

pu: for PCMU codec
pa: for PCMA codec
lb: for iLBC codec
su: for speex ultrawideband codec
sw: for speex wideband codec
sx: for speex narrowband codec
g6: for G.726 codec
gg: for GSM codec
g2: for G.722 codec

9. Copy the raw file(s) to the ConvertRaw directory
10. Run: ConvertRaw.exe “raw.EXT” “out.wav”
    (You can also combine in/out in a single stereo file: converter.raw “in.EXT” “out.EXT” “out.wav”)
11. Listen out.wav with any audio player

5.133. How to use long called prefix lists for routing

Add or import the prefixes to tb_routingprefix.
Add the following to the v_getroutingtime for the calledtechprefix selection:
OR EXISTS (select top 1 tb_routingprefix.id from tb_routingprefix WITH (NOLOCK) where tb_routingprefix.routingid = tb_routing.id and @called_norm2 like tb_routingprefix.prefix+'%')

5.134. Send email to users on low credit

The low credit notification email will be sent to prepaid users with an email address after calls when the user credit cross (goes below) the minimum credit amount.

It can be configured by the “sendlowcreditemail” global config and per user (tb_users.sendlowcreditemail).
Possible values:

-2: yes, with auto guessed minimum credit amount value
-1: auto guess if send is needed for prepaid endusers, resellers and traffic senders if the SMTP email account is configured
0: don’t send
others: specify minimum credit amount

Default value is: -1.

The default emails subject and message can be modified with the lowcreditemailsubject and lowcreditemailbody global config options.

The following keywords can be used (these will be auto replaced at runtime):
[newcredit],[currency],[username],[companyname],[callerid],[callerip],[callernumber],[callername],[calledid],[calledip],[callednumber],[webportal].
If these are not configured, then the default will be something like this:
lowcreditemailsubject:

> [servername], Low VoIP credit notification

lowcreditemailbody:

> Your VoIP credit is running low.
> Your credit is: [newcredit] [currency].
> Username: [username]
> Recharge required to continue using the service.
> You can unsubscribe or modify your settings from the web control panel. (This line is sent only if the webportal is configured)
> [companyname]

Applying:

- To disable all low credit notifications, just set the sendlowcreditemail global config to 0.
- To enable low credit notifications, set the sendlowcreditemail global config to -2 or to a positive value (which will be used also as minimum credit) and make sure that you configure email server SMTP settings (emailhost, emailuser, emailpassword global config values).
- Otherwise (if the sendlowcreditemail global config is -1 which is also the default) the server will auto guess if it needs to send notifications or not.

If the sendlowcreditemail global config is not 0 then it will be overwritten by the user sendlowcreditemail (defaults to -1 / auto guess).

- To disable low credit notification for a user, set it's sendlowcreditemail to 0.
- To enable low credit notification for a user, set it's sendlowcreditemail to -2 or to a positive value for the minimum credit.

To send emails only to a few selected users, then:

1. Configure the SMTP credentials from the Config Wizard or by the emailhost, emailuser, emailpassword global config values
2. Set the sendlowcreditemail global config to -2 or to a positive value
3. Disable email notifications for all users by default with the following query (copy-paste to the Direct Query form):
   update tb_users set sendlowcreditemail = 0 where sendlowcreditemail = -1
4. Enable email notifications for the desired users by setting their sendlowcreditemail field to -2 or to a positive value

If the SMTP credentials are not configured, then the server can use a built-in email account to send the emails, but this account is limited to a number of maximum emails.

5.135. Built-in webserver

The service has a simple built-in web service which can be used to host various files. For example it can be used to host the customized softphones and other resources.
(This is not the same as the enduser web interface. The enduser web interface is hosted in a separate process).
Just put your files to its directory (which is the "mvweb" directory by default).
From the clients just use the web directory in requests, such as http://serveraddress/mvweb/index.html

Settings:
You can set these on the "Configurations" form (global settings):

- hasbuiltinweb: -1=auto (default), 0=no (disable), 1=yes (enable), other=only on this specific port
- allowhttpfileupload: 0=no (default),1=yes (free access; used for file sharing and profile picture), 2=to specified dirs with auth, 3: both
- httpfileuploadkey: required only if the allowhttpfileupload is 2 or 3
- builtinwebdir: default path (defult is \mvweb)

Simple file upload:
Enable by setting the allowhttpfileupload global config to 1 (or higher if you need also auth).
The files will be saved as is under \mvweb\filestorage\file
Download link: http://serveraddress/mvweb/filestorage/file
URL:
http://yourserver.com/mvweb/myuploadhtml.html
or
https://yourserver.com/mvapireq/?apientry=fileupload&requestid=1
The URL path/file name must contain the "upload" string.
Client upload as simple multipart/form-data or JavaScript fetch with the file name sent by the "filename" parameter.
Parameters:

- filename: The original file name. This is sent automatically for uploads.
- fsavename: optional file name to save as. If not set, then it will use the original filename.
- filepath: optional file path with the parameter

Note: if the allowhttpfileupload is set to 1, then no authentication is performed for file transfers except file i/o rights. For private access you should set the file names accordingly (long file names generated by hashing the username+password+privatesalt)
Note2: no special characters are allowed in file names or paths. Max 4 folder level is allowed.
Note3: For smaller files the API can be also used (this can be used also with websocket, json and others):

putfile, pputfile [filename, filedata] {will save a file under the mvweb\filestorage directory. The filedata should be base64 encoded}
getfile, pgetfile [filename] {will load a file from under the mvweb\filestorage directory }

Upload voice files:
Enable by setting the allowhttpfileupload global config to 2 or 3.
The files will be saved as is under \voices\.
URL:
https://yourserver.com/mvapireq/?apientry=fileupload&authkey=apiv2key&fsavekey=httpfileuploadkey&fsavetype=voices&requestid=1
or
https://yourserver.com/mvapireq/?apientry=fileupload&requestid=1
(and pass the other parameters in the POST body)
Client upload as simple multipart/form-data or JavaScript fetch with the file name sent by the "filename" parameter.
Parameters:

- filename: The original file name. This is sent automatically for uploads.
- fsavename: optional file name to save as. If not set, then it will use the original filename.
- fsavetype: must be set to "voices"
- authkey: the apiv2key or httpapikey global config
- fsavekey: the httpfileuploadkey global config

Examples:
See the upload_html_form.html and the upload_js.html files in your server app folder \mvweb\ subfolder.

5.136. Analyzing mizu server log files

The log files are stored in the server directory below the logs folder or in a directory specified by the "logpath" global config option with "X_log_DATE_mserverdebuglog.dat" file name format.
Usually you can find the logs in the server app folder (near the mserver.exe) -> "logs" subfolder.
You can also open the folder from MManage -> Files Menu -> Folders -> Server logs directory.
You should use an app which is able to open huge files, for example F3 from Total Commander (don't use Notepad.exe to open large log files).
The amount of the logs depends on the "loglevel" global config option. 1 means minimal, 5 means details (don't set to above 5 as that is only useful for rtp debug). You can quicly set the log level from MManage -> Control menu -> Logs -> Set. "On" means loglevel 5.
Important (level 1) messages are also stored in the database (tb_logs, viewable from MManage "Logs" form).
Open the last logfile ("log_xxx.dat") with a fast text viewer (For example F3 from TotalCommander). If the problem is not in the last file, you might search for the related SIP Call-Id or "INVITE sip:number" in all files for example using Total Commander -> Commands -> Search -> Find Text. (On the "Advanced" page you should set the "Not older than" option to avoid searching across too old files.

Search for "ERROR" or "catch" to detect issues.
To be able to understand what is happening in the log, you need the following knowledge:

-protocols: IP, UDP, TCP, SIP, RTP, other related protocols

-SIP. especially you should understand the SIP registrar process and basic call flow: https://tools.ietf.org/html/rfc3665

-SQL (SELECT/INSERT/UPDATE/DELETE/stored procedures) and the structure of the mizu voip database (tb_users, tb_cdrs, tb_currentcalls, stored procedures and other tables).

In case if the logs doesn’t contain the SIP signaling details, enable the detailed logs first (Control menu -> Logs -> Set -> On), reproduce the problem and check the log files again.

For registration related problems search from @USERNAME from the bottom up to find out the SIP REGISTER messages from a user. If the user is registered, then you should see its status accordingly on the "Users and devices" form (statusdate is recent and status is 1 or higher. After the first register, the server usually sends a 401 Unauthorized answer asking for authentication. Once the user is authenticated you will see an update to tb_users in the logs (update tb_users with(ROWLOCK) set StatusDate = getdate ..)

For call related problems first find the call-leg. This can be done by searching for "INVITE SIP: callednumber" from the bottom up (if the call was made recently). Once you find this message, copy the "Call-ID" header to the clipboard and search for that id from the beginning of the log. There are 2 call-id related to each call. A-leg from client to server, b-leg from server to destination. The b-leg call-id is the same as a-leg call id with a few more characters appended, so you can easily find each-other. You might have to go through both call-id to trace a call. When the server disconnects a call for any reason, you should find related logs before or during the disconnect (and you should check the SIP disconnect code and the "Warning" header)

5.137. How to set customer disconnect reason code?

For H.323 just set the RRR_ global settings to a proper Q931 disconnect code
For SIP just set the SSS_ global settings to a proper SIP disconnect code (400-599)
The above settings will affect the disconnect reason emitted by the routing.
For disconnect reason emitted by the SIP engine, you might change the values by using the changediscreason1,changediscreason2 .. changediscreasonN global config option with a key value containing ORIGINALCODE=NEWCODE (so just separate the old from the new code with equal sign). For example to change the code 408 to 409 just set "changediscreason1" global config key to value "408=409".

For some disconnect reasons the server might play a voice announcement for the caller to notify it about the disconnect cause. You might turn this off by setting the "allowdiscmessage" to 0. You can change the voice messages by replacing the appropriate voice files or by setting the allowdiscmessage to 2 you can use separate voice files per message named as disc_XXX.wav where XXX is the disconnect reason code.
The wave files must have must be standard 8 kHz 16 bit mono PCM files (128 kbits - 15 kb/sec).

Search for "disc" in the global config for more options.

5.138. Time display

Defaults:
In the database the date-time variables are stored according to **database server time-zone** by default (the database engine time settings, which by default depends on OS time-zone and settings).

On the user interfaces the time is displayed by default according to the **app time zone**. For MManage this means the time-zone of the computer where it is running. For the webportal it means the webserver time zone.

This default time display on user interfaces can be changed by the "**timesetting**" global config option:

- 0=db server,
- 1=local (default)
- 2=UTC diff (which can be set by user)

MServer:
By default the mserver will use the same time as the database. This is the recommended setting, so you will have to change the times when displayed on user interfaces.
If the **usedbdatetime** is set to 2, then the server internally will use the same time as the database (this is the default behavior and recommended to leave it as is).
Otherwise (if you set the usedbdatetime is set to 0) it will automatically calculate the difference in time between the app server and the databse server (if these are running on different servers).
In this case you can verify the followings in the server logs:

```
EVENT, timezonediff is set to XXX (timezonediff)
```

MizuManage:
In the MizuManage you can change (overwrite) the local display settings from **Config menu -> MManage** settings.
This will affect grid display and date-time chooser controls only, so you can still edit the values in server time (it is not possible to change the time for the server).

Webportal:

For the webportal you set the default time-zone with the “**fixutcdiff**” (UTC difference in minutes from server time; Mizu hosted servers are located by default at UTC+0, UK time zone). This will be considered only if timesetting is set to 2. The users can also set their UTC time if allowed by the "feature_usertimezone" setting.

If you wish to disable/enable user controlled time zone, you can do so with the "**feature_usertimezone**" setting.

- 0: disable
- 1: allow using the user “utc” field,
- 2: allow user edit (default)
- 3: means to also auto set the default based on user country which is selected during user signup or user settings

Verify the followings in the webportal logs:

```
EVENT, timezonediff local set to XXX (srvtimezonediff)
EVENT, timezonediff fix set to XXX (fixutcdiff if timesetting is 2)
```

5.140. Main/backup database

The mizu server will automatically create a secondary database named mserver_backup. This is to keep the main database small and more responsive. At every night the mserver will add all new cdr records to the backup database.
This means that the backup database contains all cdr record until yesterday. In some special circumstances this nigthly job might be skipped so it might be possible that the backup database doesn’t have the last 1-2-3 days yet.

This data migration to the backup database can be controlled with the following global config options:

- dbmaint_backuptables: backup cdr records and other tables to xxx_backup: 0=no,1=cdrs (default),2=extended,3=all
- dbmaint_removecdrs: remove cdr records after x days (default is 365)
- dbmaint_removeother: remove other tables records after x days (default is 750)

Once these records were instered into the backup database, the service will delete old record from the main database. This means that records from the last x days (where X means the “dbmaint_removecdrs” setting) are present in both the main and the backup database.
The backup cdr (mserver_backup.tb_cdrs) table is in the exact same format like in the main database, except the following fields:
id: it is an autoincrement field in the main dataset, but it is not an autoincrement in the backup (so the id’s are also the same)
comment: the comment field is not saved in the backup database (actually this is cleared also from the main database after a few days).
You can work exactly the same with the backup cdrs like with the main cdrs. The MManage will automatically switch to use the backup database tb_cdrs if the selected date interval is too large.

Note that this is not the same with your regular database backup. Database backups have to be made periodically (hourly/daily/weekly/monthly/yearly - full backup/differencial backup) and have to be keept on a separate media (preferably at a remote location) to keep it safe agains hardware failures.

5.141. Encrypted/hashed password

To secure your user passwords even if your database is hacked, you can set the securepasswords global config option to 1 or more.
The following values are defined:

- 0: No password hashing (easy of use but not secure)
- 1: Yes with decode possibility (recommended)
- 2: Yes with no decode
- 3: Yes forced always

5.142. How to run multiple webportals

You can run multiple webportal (enduser control panel) attached to a single service instance by setting the runwebportal2, runwebportal3, runwebportal4 … runwebportalN global config options to a NODE number.
For example:

```
runwebportal2=4
runwebportal3=5
```

will run two additional webportal on node 4 and node 5.
Note that this is another way to use NODE numbers (not for a real voip app service instance but just for webportal customizations).
Each webportal can have its own custom settings (IP/domain, logo, brand and others) specific to its node by global parameters. For example:

```
cfg_port#4=8802
brandname#4=Supervoip
cfg_description#4= Supervoip web interface
loginpage_footer1#4=Any text
```

Make sure to run each on different IP or port.

You can specify different logo images with the loginimage_X.jpg or loginimage_X.gif images where X is the node number.

5.143. How to remove duplicates from the number portability table

Execute the following SQL:

```
delete from tb_portednumbers
where
tb_portednumbers.number in
(
select b.number
from tb_portednumbers b  with(nolock)
group by b.number
having count(b.number) > 1
)
and
id <> (select top 1 c.id from tb_portednumbers c where c.number = tb_portednumbers.number order by LEN(c.newdomain)
desc, c.datum desc )
```

5.144. How to black-list/white-list client devices

The following global config options can be used to block/filter certain device types:
allowua: transport level whitelist to allow only these devices and will block all others at sip transport layer level
blockua: transport level blacklist to block these devices and will allow all others at sip transport layer level. If the entry is less than 5 character length then exact match will be required. Otherwise case-insensitive substring match will be enough.
alloweua: this is applied only for endpoints with no username/password authentication (such as IP or username only authentication) and is applied at high level only
blockeua: this is applied only for endpoints with no username/password authentication (such as IP or username only authentication) and is applied at high level only

All of the above settings allows list of equipment’s separated by comma.
For example to enable only Mizu softphones, use the followings
alloweua=Mizu,mzvios,webphone,WPhone

With the bauastrict setting you can specify how strict these user-agent strings will be checked. Possible values:
-1: auto (same as 1. default)
0: relaxed (any substring)
1: relaxed (substring, ignore case)
2: strict upper/lower-case

3: strict match not substring (but ignore case)
4: strict match upper/lower-case

The blockeuawithnouseragent settings can be used to block clients without User-Agent SIP header. Possible values:
-1: auto (same as 0/no. default)
0: no (allow clients without user agent)
1: yes only on autocreate new user
2: always block clients without user-agent

5.145. How to generate CDR's

You can generate (random) CDR records for testing purposes (for example to verify pricing/billing) from both the API and command line.
API entry: **gencdr**
Parameters:

- count: number of CDR records to generate (default is 10)

- callerid: caller id (tb_users.id). note: you can just set the caller parameter to the source user name and this will be set automatically
- calledid: called id (tb_users.id). note: you can just set the called parameter to the destination user name and this will be set automatically
- caller: caller name or number. note: you can just set the callerid parameter to the source user id and this will be set automatically
- called: called name or number. note: you can just set the calledid parameter to the destination user id and this will be set automatically
- callergroup: instead of specifying callerid or caller, you can specify a group and users from the group will be selected randomly
- calledgroup: instead of specifying calledid or called, you can specify a group and users from the group will be selected randomly
- callerparent: instead of specifying callerid or caller, you can specify a parent and sub-users will be selected randomly
- calledparent: instead of specifying calledid or called, you can specify a parent and sub-users will be selected randomly

- callerip: caller IP address
- calledip: called IP address

- callednum: called number
- callednumprefix: called number prefix (default is empty)
- callednumlen: length of the called number to be randomly generated if callednum is not set
- connecttime: pdd
- duration: call duration

    - comment: to be set in the cdr comment field (for example it can be used to delete the inserted records later: delete from tb_cdrs where datum > getdate() - 3 and comment like '%mycomment%')

Notes:

- Omitted parameters (empty strings or numbers with 0 or negative values) are generated randomly except the count (which defaults to 10 if not set) and the callednumprefix (which will not be used if not set)
- It is enough to specify one from callerid/caller/callergroup/callerparent. The same for calledid/called/calledgroup/calledparent
- From the console, the parameters have to be set in the above order, listed here in a line: count,callerid,calledid,caller,called,callergroup,calledgroup,callerparent,calledparent,callerip,calledip,callednum,callednumprefix,callednumlen,connecttime,duration,comment

Examples:

- generate 20 completely random cdr records:
    - API: http://SERVERADDRESS/mvapireq/?apientry=gencdr&authkey=XXX&authid=ADMINUSER&authmd5=XXX&authsalt=XXX&count=20&now=1
    - console: gencdr,20
- generate 30 random cdr records from a specific parent user (it's subendusers) with callednumberlength set to 12:
    - API: http://SERVERADDRESS/mvapireq/?apientry=gencdr&authkey=XXX&authid=ADMINUSER&authmd5=XXX&authsalt=XXX&count=30&callerparent=CALLERPARENT&callednumlen=12&now=1
    - console: gencdr,30,,,,,,,CALLERPARENT,,,,,,12,,,
- generate 40 random cdr records for a specific caller with callednumberlength set to 12 via a specific sip server with comment set:
    - API: http://SERVERADDRESS/mvapireq/?apientry=gencdr&authkey=XXX&authid=ADMINUSER&authmd5=XXX&authsalt=XXX&count=40&caller=CALLERUSERNAME&called=SIPSERVERUSERNAME&callednumlen=12&comment=MYCOMMENT&now=1
    - console: gencdr,40,,,CALLERUSERNAME,SIPSERVERUSERNAME,,,,,,,,12,,,MYCOMMENT

5.146. Network and IP address configuration

Most of the settings can be specified from the configurtation wizard (MManage -> Config menu) and some extra configurations are available via global configuration (MManage -> Configurations form from under the “Other” tree node) or per user (“Users and devices” form or tb_users database table).

Domain:

This can be a logical SIP domain but usually and recommended to be set for a real domain (SRV and/or A record).

For a private PBX running on a fix IP this is not an important setting, howerver if your server IP might change in the future (such as migrating to another ISP) then you should setup a (sub)domain to point to your SIP server IP and annonce this to your customer instead of the IP address (so you can change the IP and/or the whole box anytime later without the need to send the new IP to your customers, so you just need to update the domain record and your users can keep their old settings).

You should always set a domain if you wish to use WebRTC (and an SSL certificate for that domain) otherwise WebRTC will not work from Chrome and Opera.

It can be set from the config wizard or via the “LocalDomain” global config option.

Bind IP:

This is the local address where the server sockets will listen. By default the server will listen on all IP’s.

If your server has multiple network interfaces or multiple IP addres, then you can specify on which IP to listen by setting the bindip configuration from the configuration wizard “Network” section or by the “bindip” global config.

There is no need to set this if your server has only one IP or if your box is used only to host the VoIP service and there is no need to run other service or multiple VoIP services.

Public IP/Preferred IP/Local IP:

Specify which IP should be used in the SIP signaling.

For example if your server is behind a NAT, but the traffic is toward to internet, then you should specify your public IP here.

Also it should be set if your server has multiple IP’s and you don’t wish to set a bind ip.

If your server has a public IP, then there is no need to set this or it can be set to the same value as the bind ip.

However if your server is on a local LAN and you wish to offer VoIP services for external network or via the internet, then you have to set this to your external public IP. This can be set with the “Public IP” setting from the configuration wizard or via the “localip” global config option.

InternalIP:

Sometime you need to use a separate IP toward your external network (such as your carrier or gateway) and internal network (for your users/extensions). In this case you might set the InternalIP global config option to be used by the endusers.

Offer services for:

This is shown if your server is on LAN.

Specify from where you will use the VoIP server. From where the clients will connect.

- Both LAN and Internet: the VoIP server will auto-detect the IP to be used for each session
- Internet only: use this if your server is on public IP and all peers are from other locations (not from local PC or local LAN). This will work as a hint for the server NAT handling algorithm, so it will never try to use private LAN IP’s in signaling

- LAN: select if all your clients are connecting from the local LAN but you also receive or send traffic from the internet (using outbound trunks)
- Force LAN always: select this if there are no any outbound trunks, so all traffic is within your LAN

Allow LAN peers:

This is shown if your server is on public IP and will help the internal NAT algorithm to decide whether to allow/try private IP's in the SIP and SDP messages or force only public IP's

- Auto (auto-detect. recommended)
- No (all peers are accessed through the public internet)
- Yes (some or all devices are on local LAN. use Auto instead)

SIP port: the main SIP signaling port (UDP and/or TCP). You should always prefer port 5060.
H.323 port: specify if you enabled the H.323 module. Usually 1720.
Access port: this port will shared among the extra services such as the web portal access, API access, TURN and others
SIP transport: You should always select UDP here. TCP and TLS are optional
Auto get SSL/TLS certificate: if you need TLS (for example for WebRTC) then the service can acquire and manage free Let's Encrypt certificate for you
Unblock windows firewall: This should be set only one time to add exceptions to the windows firewall
Enable dynamic firewall: Check this to enable the built-in protection

Different network interface (Ethernet interface) per provider:
In some situations you might have a direct peering connection with some carrier or gateway.
In this case you can set the server to listen on different interfaces by the secondarybind global configs:

secondarybindip1, secondarybindport1
secondarybindip2, secondarybindport2
…
secondarybindipN, secondarybindportN

(N is max 125).
Then you can set the "interface" field for users to a number to match one of the global config secondarybind entry.

Explicitly set the IP address used in signaling

The server might auto-detect another (best match) address to be used in the SIP signaling even if you have set a bindip and/or localip. This might be the case on servers with multiple network interface or a different/changing external IP.

You can use the following global config options to prevent this:

- autouselocalip: set to 0
- autodetectlocalip: set to 0

Additional related parameters:

- localip (you might set this explicitly)
- bindip (you might set this explicitly)
- localinternalip (you might set this explicitly)
- nathandling
- nataddrtype
- usephisicalfromaddr

You can also set a different interface per provider as described above.

Other related settings:
nathandling: wether to accept lan IP in sip signaling if bindip is public //0=old manual hints (hasinternalaccess,canacceptlocalip,denyaddr), 2=new automatic, 4=allow only public address, 6=allow all addr except self, 8=allow ALL including self address, 10=allow only lan address
nataddrtype: which of our address to use if bindip is private: 0=auto, 1=public only, 2=private only
autodetectlocalip: 0=never, 1=no,2=yes,3=always
localport: SIP listen port
localport2: additional SIP listen port
Also search for “bind”, “ip”, “port”, “nat” from the Configuration form for more settings.

5.147. Port forward for NAT

If your server is behind NAT or firewall and you wish to offer services for external networks (internet), then you should set your Public IP (localip global config) and setup symmetric port forwarding on your firewall (external and internal ports to the same forwarded to your internal box IP).
You need to forward the following ports (check them in the global configuration by searching for their names from the MManage “Configurations” form):

- Remote desktop TCP 3386 (for easy server administration)
- TCP 1433 and/or 2223 (for SQL server, otherwise you will not be able to use MManage remotely)
- localport UDP and TCP (main SIP signaling port. Default is 5060)
- localport+1 TCP (for secure SIP sips. Default is 5061)
- callcenterport UDP and TCP (only for outbound callcenters)
- callcenterport+3 and callcenterport+4 TCP (if you need remote FTP)
- mainaport UDP and TCP (main server port for various purposes including API, websocket, webportal, TURN and many others. Default is 80.)

- mainaportupd UDP (if set)
- adminport TCP (for remote CLI access from MManage server console)
- monitorport TCP (to easy access logs from remote MManage)
- ssl port TCP (443 by default for https and wss)
- MinRTP-MaxRTP range UDP (for RTP; enable also TCP if you are using WebRTC since it might be used for TCP ICE candidates)
- fs_minrtp- fs_maxrtp range UDP (RTP for WebRTC and extra PBX)
- tcpcandidatesrvport (TCP relay port, 10080 by default)
- forwarderport (TCP forwarder for various internal services, 11080 by default)
- H.323 signaling TCP (port 1720 and 1721 by default if you need H.323)
- UDP: 44444 (“voice-here” functionality in MManage)
- TCP: 9885, 9886, 9889 (optional ports for gsm server, admin port, log port)

Search for “port” in the Configuration form for more.

5.148. Autoprovisioning

All the mizu SIP clients can be completely preconfigured, so the users just have to type their credentials to begin the usage.
In case if you wish to change the settings from server side, you can use the webphone which loads all its default settings from the webphone_api.js file.
In case if you wish to use autoprovisioning for IP phones on local LAN, than you can setup a free TFTP server to store any configuration.

5.149. Disk space

Make sure to reserve enough free disk space for the softswitch. You can count with the followings:

- service binaries: around 150 MB (depending on the installed roles and features)
- database: the CDR records takes the most disk space. You can calculate with around 5MB for 1000 CDR records. These are usually migrated automatically to a mserver_backup secondary database (by default only the last 3 month records are kept in the main database to speed-up queries;)
- voice recording: it is disabled by default but it can take a lot of disk space if you enable it (it can be enabled also per user and can auto-delete old files)
- logs: the basic (1) loglevel will need only a few MB per month, however if you set the loglevel to 5 then it might need a 10-100 GB or even more if you have a lot of traffic. (The loglevel 9 can eat a lots of GB per day if your server is under traffic. This should be enabled only for short period or on test servers with low traffic). By default the last 14 days of logs are kept.

The service can manage the disk space automatically. On low disk space it will trigger warning, will lower the log levels, will delete old log files if needed and even auto temporarily disable certain features such as long comments in CDR records or call recording.

5.150. Mapping users

You can use the Mizu softswitch as an SBC to provide mapping between different credentials, so the SIP clients can use different credentials then your SIP server and the SBC will convert between these.
You need to following settings to enable this:

- autocreatereguser=0
- forwardauthpassword=4
- forwardauthentifications=(after your needs, usually 1)
- allowanonymouscaller = 0
- add upperusername and upperpassword to tb_users and v_checkuser
- check also upperusername in v_check_calleduser

5.151. What happens when user calls itself

You can configure the following global settings to define what to happen if a user calls its own username or phonenumber:
Allowselfcall: 0=no,1=yes and check for special calls (default), 2=yes
With the default value of 1, the routing will check the followings:

- if the “selfforwardto” global config option is set, then will forward the call to this number (you can set it to an IVR access number for example)
- if the extra pbx module is enabled, the router will route the call to the user voicemail box
- otherwise the routing will disconnect the call

5.152. Route calls between the local users via the upper server

When the mizu server is running in gateway mode (webrtc gateway, SBC or tunnel gateway), by default it will route calls between local users, so this calls are not forwarded to the upper server and back.
To force all calls via the upper server, set the following global config options:

- alloweuserusercalls: 0
- fs_directroutrfstofs: 0

5.153. Tunnel quality statistics

Use the following SQL to query advanced tunnel quality statistics:

```
--main statistics quick overwiev (also try to group by user)
select
--userid,
--case when method = 2 then 'oncalls' when method = 1 then 'onreg' else 'onelse' end as 'statfrom',
--case when currtransport = 0 then 'udp' when currtransport = 1 then 'udp_socks' when currtransport = 3 then 'tcp_randports' when currtransport = 4 then 'tcp_localsocks' when
currtransport = 5 then 'tcp_443or80' when currtransport = 6 then 'tcp_localproxy' when currtransport = 7 then 'tls' when currtransport = 8 then 'tcp_http' when currtransport =  9
```

```
then 'tcp_remotetunnel' when currtransport = 10 then 'tcp_socks' when currtransport = 11 then 'tcp_httpproxy' when currtransport = 13 then 'http' else 'unknown' end as
'currtransport',
--case when lastsucctransport = 0 then 'udp' when lastsucctransport = 1 then 'udp_socks' when lastsucctransport = 3 then 'tcp_randports' when lastsucctransport = 4 then
'tcp_localsocks' when lastsucctransport = 5 then 'tcp_443or80' when lastsucctransport = 6 then 'tcp_localproxy' when lastsucctransport = 7 then 'tls' when lastsucctransport = 8 then
'tcp_http' when lastsucctransport =  9 then 'tcp_remotetunnel' when lastsucctransport = 10 then 'tcp_socks' when lastsucctransport = 11 then 'tcp_httpproxy' when lastsucctransport =
13 then 'http' else 'unknown' end as 'lastsucctransport',
sum(callcount) as 'callcount', sum(allcallcount) as 'allcallcount', sum(allsucccallcount) as 'allsucccallcount', sum(plgoodcalls) as 'plgoodcalls', sum(subsnosucccall) as
'subsnosucccall',
sum(rtppacketsent) as 'rtppacketsent', sum(rtppacketrec) as 'rtppacketrec', sum(rtppacketloss) as 'rtppacketloss',
sum(case when srv_rtppacketsent >= 0 then srv_rtppacketsent else null end) as 'srv_rtppacketsent',
sum(case when srv_rtppacketrec >= 0 then srv_rtppacketrec else null end) as 'srv_rtppacketrec',
sum(case when srv_rtppacketloss >= 0 then srv_rtppacketloss else null end) as 'srv_rtppacketloss',
avg(case when srv_rtppacketlosspercent >= 0 then srv_rtppacketlosspercent else null end) as 'srv_rtppacketlosspercent',
avg(case when delay >= 0 then delay else null end) as 'avgdelay',
avg(case when qos >= 0 then qos else null end) as 'qos',
avg(statpoints) as 'statpoints'
from tb_tunnelstat with(nolock)
where
datum > getdate() - 3
--group by
--userid,
--case when method = 2 then 'oncalls' when method = 1 then 'onreg' else 'onelse' end as 'statfrom',
--case when currtransport = 0 then 'udp' when currtransport = 1 then 'udp_socks' when currtransport = 3 then 'tcp_randports' when currtransport = 4 then 'tcp_localsocks' when
currtransport = 5 then 'tcp_443or80' when currtransport = 6 then 'tcp_localproxy' when currtransport = 7 then 'tls' when currtransport = 8 then 'tcp_http' when currtransport =  9
then 'tcp_remotetunnel' when currtransport = 10 then 'tcp_socks' when currtransport = 11 then 'tcp_httpproxy' when currtransport = 13 then 'http' else 'unknown' end as
'currtransport',
--case when lastsucctransport = 0 then 'udp' when lastsucctransport = 1 then 'udp_socks' when lastsucctransport = 3 then 'tcp_randports' when lastsucctransport = 4 then
'tcp_localsocks' when lastsucctransport = 5 then 'tcp_443or80' when lastsucctransport = 6 then 'tcp_localproxy' when lastsucctransport = 7 then 'tls' when lastsucctransport = 8 then
'tcp_http' when lastsucctransport =  9 then 'tcp_remotetunnel' when lastsucctransport = 10 then 'tcp_socks' when lastsucctransport = 11 then 'tcp_httpproxy' when lastsucctransport =
13 then 'http' else 'unknown' end as 'lastsucctransport',
order by
sum(rtppacketsent)+sum(rtppacketrec)


--stats by transport protocol (be avare that stats can be destroyed by a few users)
select
--userid,
--case when method = 2 then 'oncalls' when method = 1 then 'onreg' else 'onelse' end as 'statfrom',
case when protonum = 0 then 'udp' when protonum = 1 then 'udp_socks' when protonum = 3 then 'tcp_randports' when protonum = 4 then 'tcp_localsocks' when protonum = 5 then
'tcp_443or80' when protonum = 6 then 'tcp_localproxy' when protonum = 7 then 'tls' when protonum = 8 then 'tcp_http' when protonum =  9 then 'tcp_remotetunnel' when protonum = 10
then 'tcp_socks' when protonum = 11 then 'tcp_httpproxy' when protonum = 13 then 'http' else 'unknown' end as 'transport',
sum(substryed) as 'substryed', sum(currsubstryed) as 'currsubstryed', sum(subsskipped) as 'subsskipped', sum(currsubsskipped) as 'currsubsskipped',
sum(subsfailed) as 'subsfailed', sum(currsubsfailed) as 'currsubsfailed', sum(subsworkinggood) as 'subsworkinggood', sum(subscallworkinggood) as 'subscallworkinggood',
sum(currsubsworkinggood) as 'currsubsworkinggood',
avg(case when delay >= 0 then delay else null end) as 'avgdelay',
avg(case when qos >= 0 then qos else null end) as 'qos',
avg(statpoints) as 'statpoints'
from tb_tunnelstat_proto with(nolock)
where
datum > getdate()-3
--and userid = x
group by
--userid,
--case when method = 2 then 'oncalls' when method = 1 then 'onreg' else 'onelse' end,
case when protonum = 0 then 'udp' when protonum = 1 then 'udp_socks' when protonum = 3 then 'tcp_randports' when protonum = 4 then 'tcp_localsocks' when protonum = 5 then
'tcp_443or80' when protonum = 6 then 'tcp_localproxy' when protonum = 7 then 'tls' when protonum = 8 then 'tcp_http' when protonum =  9 then 'tcp_remotetunnel' when protonum = 10
then 'tcp_socks' when protonum = 11 then 'tcp_httpproxy' when protonum = 13 then 'http' else 'unknown' end
order by
```

```sql
--userid,
sum(currsubstryed*3)+sum(substryed)+sum(currsubsskipped)+sum(currsubsfailed)+sum(currsubsworkinggood)


--stats about server/proxy used
select
--userid,
case when method = 2 then 'oncalls' when method = 1 then 'onreg' else 'onelse' end as 'statfrom',
ip,
case when type = 0 then 'srvtype_unknown' when type = 1 then 'srvtype_final' when type = 2 then 'srvtype_backup_final' when type = 3 then 'srvtype_main' when type = 4 then
'srvtype_mainalt'
when type = 5 then 'srvtype_backup'  when type = 6 then 'srvtype_jproxybuiltin'  when type = 7 then 'srvtype_jproxydynamic'  else 'srvtype_notrecognized' end as 'servertype',
sum(currenttrycount) as 'allcurrtrycount', sum(trycount) as 'alltrycount', sum(connect) as 'allconnectcount', sum(working) as 'alllworkingcount',
avg(case when delay >= 0 then delay else null end) as 'avgdelay',avg(case when qos >= 0 then qos else null end) as 'avgqos'
from tb_tunnelstat_srv with(nolock)
where datum > getdate()-3
--and userid = x
group by
--userid,
case when method = 2 then 'oncalls' when method = 1 then 'onreg' else 'onelse' end,
ip,
case when type = 0 then 'srvtype_unknown' when type = 1 then 'srvtype_final' when type = 2 then 'srvtype_backup_final' when type = 3 then 'srvtype_main' when type = 4 then
'srvtype_mainalt'
when type = 5 then 'srvtype_backup'  when type = 6 then 'srvtype_jproxybuiltin'  when type = 7 then 'srvtype_jproxydynamic'  else 'srvtype_notrecognized' end
order by
--userid,
case when method = 2 then 'oncalls' when method = 1 then 'onreg' else 'onelse' end,
ip,
case when type = 0 then 'srvtype_unknown' when type = 1 then 'srvtype_final' when type = 2 then 'srvtype_backup_final' when type = 3 then 'srvtype_main' when type = 4 then
'srvtype_mainalt'
when type = 5 then 'srvtype_backup'  when type = 6 then 'srvtype_jproxybuiltin'  when type = 7 then 'srvtype_jproxydynamic'  else 'srvtype_notrecognized' end


--list for a user (replace userid where clause)
select
case when currtransport = 0 then 'udp' when currtransport = 1 then 'udp_socks' when currtransport = 3 then 'tcp_randports' when currtransport = 4 then 'tcp_localsocks' when
currtransport = 5 then 'tcp_443or80' when currtransport = 6 then 'tcp_localproxy' when currtransport = 7 then 'tls' when currtransport = 8 then 'tcp_http' when currtransport =  9
then 'tcp_remotetunnel' when currtransport = 10 then 'tcp_socks' when currtransport = 11 then 'tcp_httpproxy' when currtransport = 13 then 'http' else 'unknown' end as
'currtransport',
*
from tb_tunnelstat with(nolock)
where userid = 10 and datum > getdate()-30  and method = 2
order by datum desc

select
case when protonum = 0 then 'udp' when protonum = 1 then 'udp_socks' when protonum = 3 then 'tcp_randports' when protonum = 4 then 'tcp_localsocks' when protonum = 5 then
'tcp_443or80' when protonum = 6 then 'tcp_localproxy' when protonum = 7 then 'tls' when protonum = 8 then 'tcp_http' when protonum =  9 then 'tcp_remotetunnel' when protonum = 10
then 'tcp_socks' when protonum = 11 then 'tcp_httpproxy' when protonum = 13 then 'http' else 'unknown' end as 'transport',
*
from tb_tunnelstat_proto with(nolock)
where userid = 10 and datum > getdate()-30  and method = 2
order by datum desc


select
case when type = 0 then 'srvtype_unknown' when type = 1 then 'srvtype_final' when type = 2 then 'srvtype_backup_final' when type = 3 then 'srvtype_main' when type = 4 then
'srvtype_mainalt'
when type = 5 then 'srvtype_backup'  when type = 6 then 'srvtype_jproxybuiltin'  when type = 7 then 'srvtype_jproxydynamic'  else 'srvtype_notrecognized' end as 'servertype',
*
from tb_tunnelstat_srv with(nolock)
where userid = 10 and datum > getdate()-30  and method = 2
order by datum desc
```

5.154. Turn the server to an open relay

You can use the following settings to blind accept all authentications, registers and calls:
openregistrar: accept blind registrations. 0=no (default),1=yes with any username/pwd,2=yes from low level
blindauthentication: 0= no (default), 1=yes
MINUSERNAMELEN set to 1
minpwdlength set to 1
strongdigestauth set to 0
enforcestrongauth set to 0
You might also create a traffic sender user with “Open Relay” authentication set
Also check the “Security” section and release other restrictions.

5.155. How to handle Tech Prefix

Tech. prefix in the Mizu VoIP server can be manipulated multiple ways and can be used for multiple purposes
You can set a Tech Prefix with the “Tech Prefix” field on “Users and devices” form or “techprefix” field in tb_users.
If you wish to use it for incoming authentication, then you must set it for the Traffic Sender. Tech prefix is often used in legacy networks (such as H.323) as an extra protection for authentication. For example when you define a Traffic Sender you can set it’s authentication as “Auth IP and Tech Prefix must match”.
If you wish to send it outbound, then you must set it for the SIP Server.
You can create rules to add/ remove/change the techprefix (using the “Rules” form).
You can define tech prefix based routing (“Routing” form).

5.156. Route incoming calls to user groups

Calls between endusers are routed automatically. However in some circumstances you might want to route incoming calls to a group of users.
This can be done in the following ways:

- Via virtual number:
  Just set the same number for the users’ “Tel Number(s)” field thus when an incoming call arrives it will find all users with the same number assigned and the server will route the call to the “best” user (available/registered, not in call)
- From routing:
  Create a routing rule where you can set the “Called prefix” to a DID number, then on the right side add a user group or separate enduser records. The call will be routed to the “best” user available in your group
- Ring Groups and Call Fork:
  As described here

The server can also automatically reroute the call if the called user is not available.
You can also setup various rules to forward the call to other users if a user is not available or doesn't pickup the call.

5.157. How to prevent SIM card blocking

Mizutech already discontinued VoIP-GSM gateway manufacturing, however the Mizu VoIP server is compatible with all SIP capable GSM gateways.
It is not possible to entirely solve the SIM blockage problem from the VoIP server itself as this can be solved only by the gateways you are using.
To prevent SIM blockage you should use some gateway's which:
-also receives (fake) in-bound calls
-sends/receives some occasional (fake) SMS
-moving (the SIM cards could be virtual routed to other locations)
-other tricks to prevent operator blockage
-enable caller banning
If your gateway is capable for this, the rest can be handled by the VoIP server (distributing the calls across the gateway, bad gateway detection, failover and others)

5.158. How to play a voice file before calls

You might play a voice/sound file(s) before calls to announce something (such as user remaining credit), advertisements or any other promts or messages for the caller user.
There are multiple ways to enable this feature:

- Globally
- Per user
- IVR

For this feature to work, the RTP streams must be routed via the server. This should be done automatically if voice playback is needed for a call or it can be set globally from Config menu -> Configurations -> RTP Relay -> Route or per user from Users and devices form -> Functions tab -> Media relay -> Always route.
This feature works if one of the call legs is on SIP.

Global configuration

The global configuration can be set from "**Config Wizard**" -> "**Routing**" page -> "**Promt before calls**" section or by modifying the related global settings on the "**Configurations**" form:

- **defadvfile**: specify the file name to be played. The sound files must be standard 8 kHz 16 bit mono PCM files (128 kbits - 15 kb/sec) and are stored in the "voices" subfolder inside the app directory.

You can copy-paste the files manually into this directory or you can upload the files with ftp and then a “syncmedia” command must be sent from the server console. Alternatively you can use the "Media Files" form in the MManage to create and upload the voice files. Also there are some special file names which can be handled by the server automatically, including the followings: [creditdurationminminpx], [credit_tospeech_ex], [rating_tospeech_ex], and [credit_or_recharge].
You can set separate file to playback per user by the user PlayAdv field (**tb_users. playadv** or **tb_users. connectvoice** will overwrite this global defadvfile setting)

- **playadvfor**: Specify to which devices (sip endpoints) to play the file: 0=nobody,1=only not mizu clients, 2=only mizu clients, 3=all clients, 4=also for did numbers, 5=also for trafficsenders
- **discadvonconnect**: Specify how to play the file: 0=start the client ep immediately but don't interrupt the advertisement on call connect, 1=start the client ep immediately but stop the playback on connect or disconnect,2=play the advertisement before b-leg call init
- **choosecodecsonplayadv**: Specify if to select final codec: 0=no,1=for mizu ios, 2=for third party, 3=for all,4=even while already in call
- **playadvifnortprouting**: Specify force routing the audio RTP when announcements are needed //0=no,1=yes (default),2=even on high load

Example configuration to play a voice for all caller users:

- defadvfile: music
- playadvfor: 3

<u>Per user configuration</u>
The per user configuration are stored in the tb_users table and it can be set by the following fields from the **Users and Devices** form for any Endusers:

- **playadv**: 0 means no, 1 means yes, 2 means yes also for IVR calls and always owerwrite the global config
- **connectvoice**: specify the file name to be played for the user. Same as the global defadvfile but applied only for the current caller user. Will owerwrite the global defadvfile if the playadv field is set to 1 or 2.

In case if you wish to play a voice only for certain users, then leave the defadvfile global setting empty and set the playadv to 1 and the connectvoice field for the required user(s).

<u>IVR configuration</u>
For more complicated announcement you can use the IVR (especially if some logic must be implemented such as deciding what/how to play based on the circumstances or if user interaction is required via DTMF).
In this case you will need to force the calls via the IVR, create any script actions, then forward the call to the initial destination number
See the IVR guide for more details.

5.159. How to enable VoIP push notifications?

The VoIP server has full support also for push notifications. This is a useful feature to improve the availability of mobile SIP clients by sending a push notification on incoming call or text message which will wake-up the client app, thus the call/chat can be delivered even if the app is closed or the device is sleeping.

Follow this documentation to enable push notification and integrate push support into your Android/iOS/Web app.

5.160. Cannot listen on port 80

You might run the following commands if the service cannot listen on port 80 or if you receive errors like “cannot bind mmaintcpserver” or “Only one usage of each socket address (protocol/network address/port) is normally permitted”:
Find what is using the port: netstat -ano | find "0:80"
Check if your web server (IIS) uses port 80 on the IP in question.
Check HTTP.sys ports: netsh http show iplisten
Set one explicite IP if needed: netsh http add iplisten ipaddress=IPADDRESS
Remove IP if needed: netsh http delete iplisten ipaddress=IPADDRESS
Check reserved ports: netsh http show urlacl
Delete reserver UR’s for port 80: netsh http delete urlacl url=URL
Stop services which are usually listening on port 80:

```
taskkill /IM Skype /F
net stop ReportServer
net stop MsDepSvc
net stop PeerDistSvc
net stop SyncShareSvc
net stop W3SVC
net stop IISADMIN
net stop http
```

removal any network filter drivers such as WinPCap

The same applies also for port 443.

Related server config:

- sockreuse
- canselfrestart
- restartatnight
- usejobgroup
- directfilelogs
- restartprocessesatnight

More details can be found here, here and here.

5.161. How to run an query in all database

In case if you have multiple VoIP database instances, sometimes it can be usefult to select or upgrade all of them at once. This can be achieved with the sp_msforeachdb procedure.

Example:

```
sp_msforeachdb 'USE ? update tb_settings set valstr = "0" where keystr = "restartatnight"'
```

or to omit system databases

```
sp_msforeachdb ' IF "?" NOT IN("master", "model", "msdb", "tempdb") BEGIN USE ? update tb_settings set valstr = "0" where keystr = "restartatnight"'
```

5.162. How to rewrite SIP messages

Use the following global config options, to manipulate the SIP signaling messages as raw strings:

sipmsgrewrite_in_from (rewrite incoming string from)
sipmsgrewrite_in_to (rewrite incoming string to)
sipmsgrewrite_in_cond (ip address of message contains)

sipmsgrewrite_out_from (rewrite outgoing string from)
sipmsgrewrite_out_to (rewrite outgoing string to)
sipmsgrewrite_out_cond (ip address of message contains)

sipmsgrewrite_out_removeheader (remove outgoing SIP header)

For each of these you can have 100 separate entries (sipmsgrewrite_out_removeheader, sipmsgrewrite_out_removeheader1, sipmsgrewrite_out_removeheader2, sipmsgrewrite_out_removeheader3, …)

5.163. How to remove the user=phone from the request URI

Use the sendsipuseruripart global config option. Possible values:

- 0=no
- 1=for telnumbers to outbound sip servers in sip header (default)
- 2=also in to header
- 3=always

Set to 0 to remove.

5.164. How to change the domain part in the From SIP header

In SIP the From header is just the logical AOR, but some systems might be sensitive in the domain sent here.
Use the usephisicalfromaddr global config option to set it after your needs. Possible values:

- 0=will use target domain
- 1=auto (default)
- 2=will use local address most of the time
- 3=always use local address

5.165. How to set the Identity header

We already discussed the Caller-ID related settings above in this document.
This FAQ point is focusing only on the Identity headers: P-Asserted-Identity, P-Preferred-Identity and Remote-Party-ID.

Enable/disable
You disable/enable sending the identity field with the outgoing calls by the **sendidentity** global config option. Possible values:

- 0=never
- 1=no, unless enforced (default)
- 2=when required (to sip servers if phone number with call init only)
- 3=always
- 4=always P-Preferred-Identity
- 5=always P-Asserted-Identity

cli: CLIR and CLIP user settings
0: forward always (forward asserted). will not hide.
1: normal handling -default
2: forward as asserted identity always (identityrewrite asserted)
3: forward as asserted identity only to trusted domains (identityrewrite asserted)
4: normal hide (no idenityrewrite forwarding)
5: force hide (no asserted identity too!). always hided
This will set also the caller-id in the From header.
(Will set the identityrewrite, callernumber and callername ep variables)

Value to be set
By default the identity number is loaded automatically and it might be the same as the caller-id sent with the From header.

You can influence the default behavior with the following fields:
Identityrwmode/identityrewrite/identityforward: user setting to replace client->callernumber. These will affect also the caller-id in the From header.

identityrwmode: specify how to set/rewrite: 0: no rewrite, 1: basic, 2: conform sip identity specification (default)
identityforward: enable forwarding these kinds of caller-id's
identityrewrite: if the caller number don't match the identityforward prefix, then we rewrite it to this number

The per user cli setting as describe above.

You can overwrite these with the **forcedcallernidentity** global config or by the caller user forcedcallernidentity field which can be set to one of the followings:
-1: default, 0=no,1=default checks,2=db username,3=db telnumber,4=bestnumber,5=first good num,6=check minassertedidentitylen,7=with phonenumber,8=anonymous, 9=from header,10=contact header,11=Remote-Party-ID header, 12=best from client, 13=best from all, 14=called username, 15=called telnumber

This forcedcallernidentity setting is applied only for the identity headers (not for the From SIP header).

Other related config options

- parseidentity (parse incoming identity SIP headers): 0=no at all, 1=normal (default),2=rewrite caller number,3=rewrite callername also
- minassertedidentitylen: default is 6 (used with forcedcallernidentity)
- cansendremotepartyid: -1: default, 0: no, 1: P-Asserted-Identity, 2: Remote-Party-ID

Examples

- To disable sending the identity SIP headers globally set the sendidentity and cansendremotepartyid global config parameters to 0
- To disable sending the identity SIP headers from a user, set it's forcedcallernidentity to 0 and cli field to 1.
- To auto guess when to send, set the sendidentity global config at 2, leave the cli user field at 1 and the forcedcallernidentity at -1.
- To always send the identity header, set the sendidentity global config to 3,4,5 or the cli field to anything except 1 and/or the forcedcallernidentity to > 0
- To send a different From/Identity headers, set the value to be sent by the From header with the replacecalleroncalls global or user setting and set the value to be sent by the Identity header by the forcedcallernidentity global or user setting.

For example if you wish to set these for the outbound cals:

- P-Asserted-Identity: to the caller number (the Caller-ID usually extracted from the received From header)
- From: to the Username field loaded from the database

then:
If you wish to set this for all calls:

- Set the sendidentity global config to 5
- Set the replacecalleroncalls global config to 14
- Set the forcedcallernidentity global config to 12

For certain users/servers only:

- Set the sendidentity global config to 1 or 2
- Set the user replacecalleroncalls field to 14 (for the SIP Server user record)
- Set the user forcedcallernidentity field to 12 (for the SIP Server user record)

See the Caller ID config options for more details.

5.166. How to enable WebRTC

If you are using the Mizu SIP server and the WebRTC module was not enabled yet, you can enable it easily by just rerunning the Configuration Wizard (from the Config menu), selecting the WebRTC option on the Roles and Features page.

Optionally you can configure it also manually with the following global config options:

- haswebrtc: 1
- fs_use: 1
- haswebsocket: 1
- hasturn: 1
- localdomain: set to a valid domain name with its A record set to the server IP address
- autotlscert: 1
- Optional settings:
  - hasflash: 1
  - fs_pbx: 1
  - enableconferencerooms: 1
  - v2mode: 2
  - usemainaport: true
  - usetlsforall: true
  - usetlsforweb: true
  - sslportmain: 443
  - fs_restrictsiptolocal: 1
  - fs_checkdomainmismatch: 1

Switch to gateway mode (this will change the softswitch to gateway!):

- forwardauthentifications: 1

- autocreatereguser: 1
- fwdregistrations: 2
- forwardauthpassword: 1

- Optional:
  - hasbilling: 0
  - blocknotbilledcalls: 0
  - fwdunknownheaders: 2
  - fwdregistrations_xxx
  - allowupperserverselection: 1
  - runwebportal: 0
  - fastauth: 0
  - enforcestrongauth: false
  - MINUSERNAMELEN: 2
  - minpwdlength: 2
  - strongdigestauth: 0
  - allowanonymouscaller: true
  - blocksatellitecalls: false
  - blockpremiumnumbers: 0
  - builddynamicblacklist: 0
  - anumberhandling: 0
  - enablejsonp: 2
  - maxreroute: 0
  - normalizenumbers: 0
  - normalizedef: 0
  - validateinput: 2
  - normalize_clean: 1
  - allowdiscmessage: false
  - publicservices: 1
  - dbmaint_backuplevel: 0
  - dbmaint_backuptables: 0

    - playadvfor: 0

More details can be found here.

5.167. Configure SIP Load balancer

Apply the following settings if you wish to use the Mizutech SIP load balancer (MLB):

- Set alg_mizu in mlb.ini. alg_insertvia and alg_insertrr can be left empty or set to 0.
- Create separate mizu app server instances and set a different NODE number in their mserver.ini files
- Set the "mainnode" global config option to a node number. That will become the "main" server. If the main server is out, then change it to another node (This is required only if the main node is other then node 1 because 1 is the default value for the “mainnode”)
- Set the mlbhandling global config option to 1
- Optionally you might search for "mlb" in the global config and set other values such as the "mlbdomain" (your mlb server domain name)

5.168. DTMF triggered actions

The Mizu server has built-in functions to be triggered by DTMF digits, useful especially for SIP endpoints with no conference or transfer support.
Here are the defined keys:

- create conference or add new user to conference: *1*number#
- unattended transfer: *9*number#
- transfer with consultation : *8*number#
- talk with new client while in transfer: 1
- talk with old client while in transfer: 3
- disconnecting last conference party: *5*
- disconnecting conference party: *6*number#

5.169. How to backup database to a network drive

The easiest way to create backups on remote machines (network shared folders) is the followings:

1) Share the folder on the remote machine and allow access to the Administrator user
2) Use the same Administrator password for both the local machine (running the SQL server) and the remote machine
3) Set the sql service account and/or the sql server agent NT service account (LogOn As) to Administrator
4) Use the network path (UNC path) in the backup command instead of the mapped drive name. Example:
   BACKUP DATABASE [mserver] TO DISK = N'\\192.168.1.8\sharename\path\mserver_test.bak' WITH COPY_ONLY, NOFORMAT, NOINIT, NAME = N'mserver_test', SKIP, REWIND, NOUNLOAD, STATS = 10

The above method is very easy to setup but it has some disadvantages such as running the sql server under the Administrator account and using the same password on both servers, thus we recommend it only for quick tests.

You can also use different credential like this:

```
EXEC xp_cmdshell 'net use H: \\computername\sharename password/user:domain\username /persistent:yes'
```

You can also use third-party tools such as SQL Backup and FTP to upload the backups to any FTP site.

5.170. Cannot connect via RDP

Windows patch CVE-2018-0886 might cause RDP connectivity issues.
Solution can be found here.

5.171. How to block any client on a particular destination

There are two easy ways to reject calls:

- You can create a blacklist entry on the “Access Lists” form. Set the telnumber field to any number or prefix and set the calleruser field to the ID of the client (id from tb_users as listed on the “Users and devices” form)
- Or you can create a routing pattern to match that client with no destination entries (empty right-side list)

5.172. Username not accepted

The username format related restrictions are there for the following reasons:

- General security: our server has a filter for all user input and for any data that comes from outside. This is an extra security layer to avoid SQL injection and XSS attacks (in case if the data might be presented on some web interface)
- Username security: such as minimum length
- SIP protocol limitations: usernames with keywords used in the SIP protocol are not allowed (such as @, “,’, <, >, , ;, &) and some SIP devices will allow only ASCI characters

These restrictions can be influenced with the following global config settings:

- minusernamelen: minimum username length
- normalizedef: -1 default (no change), 0=no (dangerous),1=minimal (still dangerous),2=basic,3=normal (recommended),4=strict,5=extra
- validateinput: 0=no,1=basic without sql check,2=basic with sql, 3=normal,4=more,5=extreme
- strongdigestauth: -1=accept also case insensitive,0=no,1=yes (default. check nonce and realm), 2=strongest (check old nonce. not needed)

We recommend to enforce strings with at least 4 character length, ASCII only (character code between 32 and 127 decimal).

More exactly we recommend to enable only the following characters:

- a-z
- A-Z
- 0-9
- + - _ * # $

5.175. Block caller number

There are multiple ways to block a caller name/number/CLI:

- Using the firewall: Add the number to firewall list with U: prefix. For example to block user hacker, enter U:hacker as source. This is the most performant method to block a CLI as it is checked on transport level, but there might be not even logs or answers generated and will block all messages with this user in the From field..
- Blocking the user: Create a user (Users and devices form -> Enduser) with the same name/number and set it to disabled or temporary disabled
- Block from Rules: create a rule which will catch the caller number and set the “Data to set” to “Disconnect”
- Using the autoban module: Use manual caller banning: set the banning global config parameter to 1 (manual) and add the user to tb_callerstats with the good field set to 1. The autoban is only for outbound calls and will not be checked if the target user is local.
- Bock from Routing: create a routing rule which will catch the caller number and don’t set any direction for it (empty right side list and “Can Failover” unchecked)

5.176. Caller banning

This Fail2Ban module can be used to block system wide unsolicited or telemarketing calls by rejecting the caller when bad statistics are detected.
Caller banning will block the source user. In case if you are looking to block the called number in specific circumstances, look at access list instead.
To enable auto banning just set the autoban global config parameter to 2 or 3.
Once enabled, statistics will be calculated per caller number, ip (most relaxed) and number + ip (most strict).
The most important configuration parameters are autoban_minasr and autoban_minacd. Other options are set for optimal default values but can be used to fine-tune auto banning.

Configuration options (MManage -> Config -> Configurations -> Security -> Auto ban):

- autoban: enable/disable the autoban module. Possible values: 0: no/disable, 1: manual only, 2: yes/enable (might disable on high load if not effective), 3: yes/always. Default is 0. Set to 2 or 3 to enable.
- autoban_minasr: minimum asr to block per user_ip (for others it will be higher depending on the below settings). Default is 10.
- autoban_minacd: minimum asr to block per user_ip (for others it will be higher depending on the below settings). Default is 15.
- autoban_goodstatadd: will mark as good route if stats above autoban_minasr+autoban_goodstatadd and autoban_minacd+autoban_goodstatadd*2. Default is 5.
- autoban_applydays: block/allow for how many days on bad statistics. Default is -1 which means auto and will result to around 30 depending on load.

- autoban_statmincalls: nr of calls to consider for the above asr/acd statistics. Default is -1 which means auto calculated and will result to ~150 calls by default with normal traffic load.
- autoban_statdays: days for statitisics. Default is -1 which means auto calculated and will result to ~90 days by default with normal traffic load.
- autoban_numlength: check only last autoban_numlength digits for numbers to skip various national/international prefixes (change also RIGHT in v_selpatter2 if using autoban_special) . Default is 9.
- autoban_peruser_calls_multiplier: modify autoban_statmincalls for users. Default is 3.
- autoban_peruser_stat_multiplier: modify autoban_minasr/autoban_minacd for users. Default is 0.9.
- autoban_perip_calls_multiplier: modify autoban_statmincalls for ip's. . Default is 6.
- autoban_perip_stat_multiplier: modify autoban_minasr/autoban_minacd for ip's. Default is 0.7.
- autoban_pertr_calls_multiplier: modify autoban_statmincalls for traffic senders. Default is 4.
- autoban_pertr_stat_multiplier: modify autoban_minasr/autoban_minacd for traffic senders. Default is 0.5
- autoban_askkey: ask dtmf insteaf of ban. -1: auto (auto, defaults to no), 0: no, 1: yes for banned calls, 2: also for non definitive answers. If yes, then will deactivate user_ip and user_web checkings. Default is -1.
- autoban_special_voicefile: used only if autoban_askkey is set and if specified, then ask for dtmf instead of blocking. default is "press_a_key_to_continue,please_press_a_key"
- autoban_special_expecteddtmf: used if autoban_special_voicefile is set. Defauly is Y (Y means any, R means random, other means exact dtmf digits). Default is Y.

Caller auto ban works in the following ways:

1. Everything is handled at routing time using the tb_callerstats table.
2. If caller user/number/user+number is not already in tb_callerstats then the past statistics are calculated and added with the "good" field set accordingly.
    a. If the good field is 2 or more: means caller is allowed, routing will move forward
    b. If the good field is 0: means caller is blocked
    c. If the good field is 1 or null: means caller is temporarily allowed but statistics are recalculated after some calls (pendincallcount field increased)
3. The caller ban is checked only if the caller IP is not trusted and only for outband calls.

5.177. Caller banning to specific directions

This feature can be used to block unsolicited or telemarketing calls to a specific destination by blocking or rerouting the caller when bad statistics are detected.

Instead of the above described system wide banning, this option can be used to filter out traffic from sensitive directions such as a gateway which doesn't allow callcenter type traffic.

The followings are required to enable auto banning for specific directions:

- Set the autoban_special global config parameter to 1.
- Modify the v_selpattern2 stored procedure
- Auto generate tb_callerstats from scheduled tasks

The auto ban to specific directions works in the followings ways:

1. tb_callerstats table is generated from scheduled tasks
2. tb_callerstats is queried by v_selpattern2 which can skip the direction if good is 0, allow is good is 2 or more or return as isgoodcaller field if 1 or null.
3. During the routing procedure the server will ask dtmf (or reject the call) if the isgoodcaller field returned by v_selpattern2 if 1 or null.
4. If caller enters correct input then it is stored in the tb_callerstats as good 4 and b call leg is started. Otherwise if caller enters wrong or no key then it is stored in the tb_callerstats as good 0 and the call is rerouted (will call routing again with remembered alreadyroutedto).

Related global configuration options:

- autoban_special: Default is 0. Set to 1 for using call filtering per direction (requires v_selpatern2 modifications and scheduled tasks for tb_callerstats calcultation)
- autoban_special_voicefile: if specified, then ask for dtmf instead of blocking. default is “press_a_key_to_continue,please_press_a_key”.
  The server will wait for the dtmf until the playback of this file finshes, thus you might add extra silence at the end.
- autoban_special_expecteddtmf: Specify the expected DTMF key(s). Y means any, R means random, other means exact dtmf digits). Default is Y.
- autoban_numlength: check only last autoban_numlength digits for numbers to skip various national/international prefixes (change also RIGHT in v_selpatter2 if using autoban_special). Default is 9.

The tb_callerstats table must be created:

```
CREATE TABLE [tb_callerstats](
        [caller] [varchar](90) NOT NULL,
        [good] [tinyint] NULL,
        [pendingcallcount] [int] NULL,
        [expiredt] [datetime] NULL
) ON [PRIMARY]
GO

ALTER TABLE [tb_callerstats] ADD  CONSTRAINT [DF_tb_callerstats_pendingcallcount]  DEFAULT ((0)) FOR [pendingcallcount]
GO

ALTER TABLE [tb_callerstats] ADD  CONSTRAINT [DF_tb_callerstats_expiredt]  DEFAULT (getdate()+(90)) FOR [expiredt]
GO

CREATE UNIQUE NONCLUSTERED INDEX [tb_callerstats_caller] ON [dbo].[tb_callerstats] ([caller] ASC ) WITH (IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE =
OFF)
GO
```

For **v_selpattern** you will need to modify the select and the where clause.

- The select clause must return also an isgoodcaller as return 2 or more for good numbers, 0 for bad numbers and 1 or NULL for unknowns:
  ```
  ,isgoodcaller = (case when tb_routinglist.dirid <> 777 then 6 else (select top 1 good from tb_callerstats where
  RIGHT(@callernum,9) = tb_callerstats.caller) end)
  ```
- In the where clause you might add something like this to block bad numbers:

```
    AND (case when tb_routinglist.dirid <> 777 then 6 else ISNULL((select top 1 good from tb_callerstats where RIGHT(@callernum,9) =
    tb_callerstats.caller),1) end) <> 0
```

On “bad” callers if there are more SIP server entries in your routing list, then the call can failover to the next route; otherwise the call will be disconnected. Change the `777` to the ID (tb_users.id) of the target SIP server. In case if filtering is needed to multiple direction, then use something like `tb_routinglist.dirid not in (777,888,999)` where `777,888,999` are the user ID’s.

To generate the tb_callerstats table, add an “exec (update)” **scheduled task** to be run “at first start” and/or “daily” which should look like this:

```
delete from tb_callerstats

insert into tb_callerstats (caller, good)
SELECT
RIGHT(callernumber,9),2
FROM
tb_cdrs WITH(NOLOCK)
WHERE
LEN(callernumber) > 5
GROUP BY
RIGHT(callernumber,9)
HAVING
(count(id) >= 10 AND sum(SIGN(realduration))*100/count(id) > 20 AND avg(case when realduration > 0 then realduration else null end) > 30) OR
(count(id) > 0 AND count(id) < 10 AND sum(SIGN(realduration))*100/count(id) > 30 AND avg(case when realduration > 0 then realduration else
null end) > 40)

insert into tb_callerstats (caller, good)
SELECT
RIGHT(callernumber,9),0
FROM
tb_cdrs WITH(NOLOCK)
WHERE
LEN(callernumber) > 5
GROUP BY
RIGHT(callernumber,9)
HAVING
(count(id) >= 10 AND (sum(SIGN(realduration))*100/count(id) < 15 OR avg(case when realduration > 0 then realduration else null end) < 40))
(count(id) > 3 AND count(id) < 10 AND (sum(SIGN(realduration))*100/count(id) < 8 OR avg(case when realduration > 0 then realduration
else null end) < 10))
```

The above SQL will mark “good” callers as 2, “bad” callers as 0 and for the rest (non decided) the system will ask dtmf digit.
More exactly it will:

- mark callers as “good” if ASR is above 20 and ACD is above 30 if caller had at least 10 calls.
- mark callers as “good” if ASR is above 30 and ACD is above 40 if caller had less than 10 calls (so even after a single call)

- mark callers as “bad” if ASR is below 10 or ACD is below 20 if caller had at least 3 calls.

In case if the caller is neither “good” or “bad” then dtmf key can be asked.

- If the caller types the expected dtmf, then it will be added to tb_callerstats at runtime with good set to 4.
- If the caller doesn’t type any key or enters a wrong value then it will be added to tb_callerstats at runtime with good set to 0.

Notes:
If your server is set to autorestart at night on no traffic, then usually it is enough if the tb_callerstats scheduled tasks is set to run only at first start.
Even if caller enters a wrong key, it will not be blocked for forever. After some time the system will ask key again as the statistics window expires.
Even if caller enters a good key, it might be blocked later if it begins to send bad traffic.
If you enable banning to specific directions to some server(s) only, then in the routing you might add secondary routes where the call can be routed if the actual caller was banned.

5.178. Fine-tune caller banning to specific directions

Once caller banning for specific directions are enabled, you might fine-tune it after your needs:

- In case if you don’t want to ask dtmf, just block or reroute unknown numbers, then set the autoban_askkey global config option to 0 and modify the where clause of v_selpattern2 stored procedure to:
  ```
  AND (case when tb_routinglist.dirid <> 777 then 6 else ISNULL((select top 1 good from tb_callerstats where RIGHT(@caller,9) =
  tb_callerstats.caller),0) end) > 1
  ```
- In case if you wish to allow also unknown numbers (and block only bad numbers without asking for any dtmf key), then set the autoban_askkey to 0 and modify v_selpattern2 select clause like this:
  ```
  ,isgoodcaller = (case when tb_routinglist.dirid <> 777 then 6 else when ISNULL( (select top 1 good from tb_callerstats where
  RIGHT(@caller,9) = tb_callerstats.caller),1) = 1 then 2 else (select top 1 good from tb_callerstats where RIGHT(@caller,9) =
  tb_callerstats.caller) end)
  ```
- In case if filtering is needed to multiple directions, then in v_selpattern2 use something like `tb_routinglist.dirid not in (777,888,999)` where `777,888,999` are the SIP server ID’s.
- In case if you wish to change the playback voice, upload any voice file and set the autoban_special_voicefile global config option to your file name.
- To change the minimum number of calls to consider for “bad” statistics, change the `count(id) > 10` in the build callerstats scheduled task to `count(id) > x` (where x is the minum calls to be used such as 20).
- To change the minimum ASR to block/allow, change the values in the build callerstats scheduled task for `sum(SIGN(realduration))*100/count(id)` calculation.
- To change the minimum ACD to block/allow, change the values in the build callerstats scheduled task for `avg(case when realduration > 0 then realduration else null end)` calculation.
- To list all caller banning statistics, use the following SQL:
  ```
  select * from tb_callerstats
  ```
  (good: 0 means bad, 1 or null means no definite answer (probably new caller), 2 means good,3=means very good,4=key accepted

- To check if a number is on the list, use the following SQL:

  ```
  select top 1 good from tb_callerstats where RIGHT('80670100',9) = tb_callerstats.caller
  ```

  Replace 9 with the autoban_numlength config value if it have been modified from the default 9.
- To delete all previous statistics:

  ```
  delete from tb_callerstats
  ```

  Warning: this will delete all previous statistics!
- To delete all banned numbers:

  ```
  delete from tb_callerstats where good < 1
  ```

  Warning: this will re-enable and allow all numbers!

5.179. How to generate reports for traffic senders

You can generate reports with statistics and CDR's for any users.
Just set the senddailyemail and/or sendmonthlyemail user field to 1 or higher. Possible values: 0: no, 1: yes, 2: include also CDR's, 3: save only to file.
If set to 3, then you will find the reports in the reports subfolder.
Related global settings:

- sendreports: 0: no, 1: only if there was activity (default), 2: always, 3: force
- keepreports: number of days to keep the files (default is 14)
- cdrreporttype: -1: auto (default; connected only), 0: connected only, 1: all

Make sure that a correct SMTP email account is configured, otherwise the built-in account will not work after some time.

5.180. Music on Hold

Music on hold (MOH) can be enabled/disabled from the Configuration Wizard -> Other settings -> MOH (checked means enable, grayed means auto/default, unchecked means disable).

Music on hold can be also configured with the following global settings:

- holdmusic: Music file. 0/false: disable, empty/null/1/true: default, other: set
- musiconhold: Specify if music on hold should be played. -2: don't change, -1: auto/default (not for SIP, keep fs as-is), 0: no (disable also for WebRTC), 1: if RTP is routed, 2: always, 3: also on high load
- musiconholdat: Specify when to play music on hold. -1: auto (0), 0: on sendonly received from the other party, 1: on recvonly (default), 2: on recvonly forwarded from the other ep (default), 3: always, 4: force always (even if there is ivr)
- musiconholdwhen: Specify when to check music on hold playback. -1: auto (same as 2), 0: on sendonly receive from the other ep, 1: on inactive received (will change to sendonly), 2: on both
- musiconholdforce: Force music on hold. -1: auto (same as 1). 0: no, 1: yes (suppress audio from the other ep on MOH)
- playtransfertone: Enable/disable ring or music playback on transfer. 0: nothing, 1: ring, 2: music (default)

You can also change the default music to be played for SIP by changing the music.wav file (8 kHz 16 bit mono PCM in the app folder "voices" subfolder - MManage -> File menu -> Folders -> Voices directory).
For WebRTC the music files are located at \tsmodules\fs\sounds\music\.

5.181. Pause between calls

You can set the mincallwait field for the users (tb_users.mincallwait in seconds) to specify a minimum wait time to the same destination.
This will cause the routing (v_check_calleduser, v_selpattern2) to check the seconds elapsed since lastcalleddt.
Related global setting: updateusrcalldate (specify when to upgrade last call date-time: -1: no even the lastcalled, 0: no, 1: only for connected calls/default, 2: always, 3: also at call begin/routing)
Related fields: lastcalled, firstcallerdt, firstcalleddt, lastcallerdt, lastcalleddt

5.182. Troubleshooting

The Mizu VoIP platform provides a rich set of tools that you can use for troubleshooting general issue, register issues, call issues or other problems.

You can use the following tools from MManage:

- Have a look at the “Dashboard”
- Have a look at the “Analyze” for to detect any potential issues
- Check any important errors or warnings on the “Logs” form
- If you are making call to a local user, make sure that the called user is registered when you call it (displayed on the “Users and devices” form)
- In case of call failure you can check the disconnect reason from the “CDR” form
- If the problem is not obvious using the above tools, then you should look in the server logs.
    - Make sure that logs are turned on: MManage -> Control menu -> Logs -> Set -> On
    - Verify the log files. You can find the logs in the server app folder (near the mserver.exe) -> “logs” subfolder. You can also open the folder from MManage -> Files Menu -> Folders -> Server logs directory.
    - Open the last logfile (“log_xxx.dat”) with a fast text viewer (For example F3 from TotalCommander). To find application errors, open the last log file in the server app directory and search for “ERROR” and “WARNING”. To find a call, search for “INVITE sip:callednumber”.
    - More details about working with log files can be found here
- If the problem is not caused by the Mizu server, check the logs of the peers (softphone logs, webphone browser content, remote Asterisk logs, etc)

5.183. How to get support

Mizutech will require a remote desktop access to your server to be able to provide proper support.

Make sure that your server meets the minimal hardware requirements and network bandwidth eligible for remote administration.

With any support request please send a detailed and precise description of the problem with the exact details and a detailed log.
One of the followings will be required:

- detailed log from server and/or client side (in case if you are using a SIP client that can generate logs. Just make sure that you are using a high log/debug level so we can see also the SIP signaling in the log)
- or a network trace (such as a wireshark pcap file)
- or a SIP Call-ID (the value of the Call-ID SIP header of the failed session)
- or a CDR ID (this can be found in the MizuManage CDR form. If you make a call and open the CDR form with no any filters then the your last call will be the first in the grid. Send it's id field or right click to grab all the details)
- or call details (only if the above are not possible):
  - caller number
  - called number
  - date-time when the call (after the server time / time-zone)

Working with log files

To understand the log files, you need some basic SIP protocol understanding. Here are some basic message flow examples.
In case if the softphone doesn't connect or register, look for WebSocket (ws/wss) connectivity issues or register issues (search for "REGISTER sip").
In case if there is some call related problem, find the call in the log (search for "INVITE sip") and then go trough the call by searching after the Call-ID field what you found in the first INVITE.
If the problem is not obvious, send the log to mizutech support.

6. Links

VoIP Server homepage: https://www.mizu-voip.com/Software/VoIPServer.aspx
Mizutech homepage: https://www.mizu-voip.com/
Documentations: https://www.mizu-voip.com/Support/Documentations.aspx
Email support: serversupport@mizu-voip.com