

Copyright notice & disclaimers

Copyright (c) 2001-2006 PortaOne, Inc. All rights reserved.

**PortaBilling100 & MVTS integration guide, August 2006
Maintenance Release 12
V1.12.7**

Please address your comments and suggestions to: Sales Department,
PortaOne, Inc. Suite #400, 2963 Glen Drive, Coquitlam BC V3B 2P7
Canada.

Changes may be made periodically to the information in this publication. Such changes will be incorporated in new editions of the guide. The software described in this document is furnished under a license agreement, and may be used or copied only in accordance with the terms thereof. It is against the law to copy the software on any other medium, except as specifically provided in the license agreement. The licensee may make one copy of the software for backup purposes. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopied, recorded or otherwise, without the prior written permission of PortaOne Inc.

The software license and limited warranty for the accompanying products are set forth in the information packet supplied with the product, and are incorporated herein by this reference. If you cannot locate the software license, contact your PortaOne representative for a copy.

All product names mentioned in this manual are for identification purposes only, and are either trademarks or registered trademarks of their respective owners.

Table of Contents

Preface 3

1. Solution Architecture.....4

 Concepts..... 5

2. MVTS Installation and Configuration7

 Installing MVTS..... 8

 Configuring MVTS..... 9

 Testing the MVTS configuration..... 11

3. PortaBilling Configuration 13

 PortaBilling100 Configuration..... 14

 Call Routing 34

4. SIP-HIT Configuration 39

 SIP-HIT & MVTS Configuration 40

 PortaBilling100 Configuration for SIP Routing via MVTS..... 42

5. How to..... 45

 Configure Netmeeting to make calls via MVTS 46

 Terminate calls to my partner via his gatekeeper..... 47

6. Troubleshooting..... 49

Preface

This document provides information on setting up PortaBilling100 in conjunction with MVTS to provide various VoIP services, such as H323 traffic exchange.

Where to get the latest version of this guide

The hard copy of this guide is updated at major releases only, and does not always contain the latest material on enhancements occurring between minor releases. The online copy of this guide is always up to date, and integrates the latest changes to the product. You can access the latest copy of this guide at: www.portaone.com/resources/documentation/

Conventions

This publication uses the following conventions:

- Commands and keywords are given in **boldface**
- Terminal sessions, console screens, or system file names are displayed in fixed width font



Caution indicates that the described action might result in program malfunction or data loss.

NOTE: Notes contain helpful suggestions about or references to materials not contained in this manual.



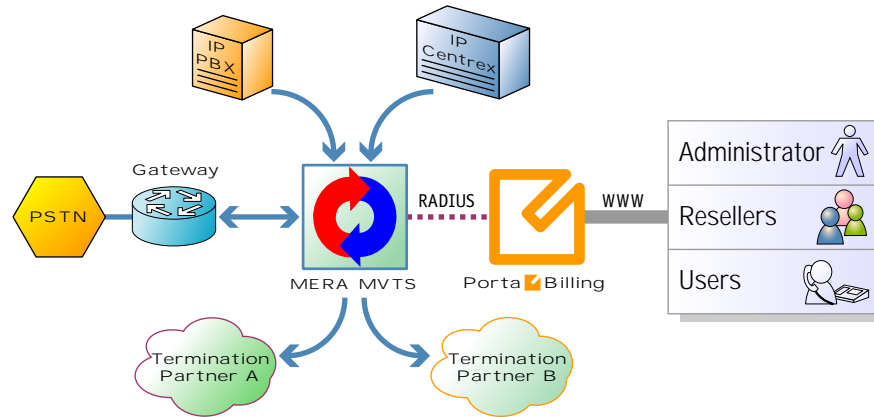
Timesaver means that you can save time by performing the action described in the paragraph.



Tips provide information that might help you solve a problem.

1. Solution Architecture

Concepts



VoIP nodes

A node is an element of your VoIP network, such as a gateway, proxy, gatekeeper, and so on. A node:

- participates in the call flow (passes a call through itself),
- communicates with the billing in order to determine whether a particular call should be allowed or not,
- provides accounting information to the billing after the call is completed, so the call can be charged.

MVTS is a typical VoIP node: it accepts incoming calls, performs authorization in the billing, sends calls to the remote gateway for termination and, finally, delivers accounting records to the billing.

RADIUS server

In order to communicate with external entities such as VoIP gateways, the RADIUS protocol is used. VoIP nodes serve as radius clients, and PortaBilling100 serves as a radius server. The client sends a request (set of attributes) to the server, and the server replies. There are three types of requests:

1. **Authentication** – the VoIP node have to determine whether a certain account (the account ID passed in the User-Name attribute) is valid and is allowed to use the service. Billing replies with either a rejection or an acknowledgement (including optional attributes such as h323-credit-amount – available funds).
2. **Authorization** – the VoIP node have to determine whether an account is allowed to make a call to a specific destination. Billing replies with either a rejection or an acknowledgement (including

optional attributes such as h323-credit-time – maximum allowed call duration).

3. **Accounting** – the VoIP node sends information about the completed call, and the billing replies with a confirmation that it has been processed successfully.

MVTS

MVTS is an advanced H323 softswitch incorporating the H323 gatekeeper and proxy functionality (with the ability to perform both signaling-only and media proxying).

MVTS-PortaBilling100 dialogue

MVTS sends requests to the billing so as to obtain information for the following actions:

- **Registration.** A new H323 endpoint tries to register to the gatekeeper. MVTS sends an authentication request, and the endpoint is registered upon successful authentication.
- **Call authorization.** One of the endpoints tries to make an outgoing call. MVTS sends an authorization request to the billing in order to check whether this account is allowed to make a call to that destination and what is the maximum allowed credit time.
- **Routing.** (This is usually done as part of call authorization). MVTS requests that the billing calculate the optimum potential routes for this call (taking into consideration cost, preferences and other carrier parameters).
- **Accounting.** When the call is completed (successfully or not), MVTS sends an accounting request to the billing, so that the account can be charged for this call.

2. MVTS Installation and Configuration

Installing MVTS

This chapter provides only a brief overview of the installation process. Please consult the [MVTS Administrator Guide](#) for more detailed instructions.

To install the MVTS application, follow the instructions below:

Unpack the archive

1. Log on to the system using the root account.
2. Copy the archive (i.e. the .tgz file – MVTS-2110-FreeBSD-eng.tgz or MVTS-2110-FreeBSD-rus.tgz) to /usr/local or any other directory of your choice.
3. Type and run the following command to unpack the archive file containing the MVTS application:

```
tar xvzf [filepath_to_archive]/[file_name]
```

If the archive is in your current directory, the file path to the archive may be omitted.

Example:

```
# cd /usr/local
# tar xzvf MVTS-2110-FreeBSD-eng.tgz
```

This will create a MVTS subdirectory (/usr/local/MVTS in our example). We will refer to this location as the **MVTS base directory** in the text below.

Set up the application

1. Change the path to the MVTS subdirectory:

```
# cd ./MVTS
```

2. Type the `sh setup.sh` command to run the setup.sh script. You will be prompted to enter IDs for the admin, support and billing groups. You may use arbitrary alphanumeric strings assigned to other groups elsewhere. (A list of groups and their assigned IDs is contained in the /etc/group file). If installation is successful, the system will respond with this message:

```
Setup finished successfully
```

Configuring MVTS

Meraproxy.cfg

Edit the `meraproxy.cfg` file (located in the MVTS base directory – `/usr/local/MVTS` in our example), section `[Radius]`:

```
[Radius]
local_address=<IP address of MVTS server>
auth_enable=1
acct_enable=1
acct_type=1
acct_leg_type=3
acct_stop_only=1
auth_password_type=0
secret=<Radius key for MVTS node in PortaBilling100>
auth_address=<IP address of PortaBilling100 master server>
auth_port=1812
acct_address=<IP address of PortaBilling100 master server>
acct_port=1813
dst_user_orig_leg=1
local_auth_port=11812
local_acct_port=11813
route_address=<IP address of PortaBilling100 master server>
route_enable=1
route_port=1812
use_h323_ivr_in=1
```

User.cfg

Edit the `user.cfg` file (located in the MVTS base directory – `/usr/local/MVTS` in our example). Change the `[default]` section according to the instructions below, and remove all other sections.

```
[default]
user=default
acct_enable=1
auth_enable=1
proxy_type=1
radius_gw_id=mvts-ua
```

Dialpeer.cfg

Edit the `dialpeer.cfg` file (located in the MVTS base directory – `/usr/local/MVTS` in our example). For external routing, only one section needs to be added:

```
[DP0]
dst_pattern=[0-9]*
gateway=EXTERNAL
priority=100
```

This will activate routing for all outgoing calls, based on the response from the billing.

Gateway.cfg

Edit the `gateway.cfg` file (located in the MVTS base directory – `/usr/local/MVTS` in our example). You must create a default section, PORTA. By default, routing replies from PortaBilling100 will have this name, so they will be processed according to configuration parameters in this section:

```
[PORTA]
address=<IP address of PortaBilling100 master server>
port=1720
gateway_mode=3
gateway_type=1
proxy_type=1
```

The **address** and **port** parameters are mandatory, so you must enter some values here – this is why we suggest entering the IP address of PortaBilling master server. However, during real call processing PortaBilling100 will provide the actual values for these parameters, so the values you enter in the file will be ignored. Thus, you may enter the IP address of the PortaBilling100 server and port 1720 (the default H323 port).

Support for PC-to-PC calls

There is a new PortaBilling feature that allows you to bill calls between end-points registered to MVTS. (This is very similar to billing SIP-to-SIP calls with PortaSIP). An example call flow is shown below:

```
NetMeeting1.[number 54321] ----MVTS---NetMeeting2.[number 12345]
```

MVTS does not provide a special indication as to whether the call was completed to one of the registered end-points. Therefore, in order to identify such PC-to-PC calls, a special prefix must be used (so that all calls to “internal” phone numbers will have a destination number starting with the same sequence of digits).

Example

You want to call Netmeeting2. When dialing 12345, you will need to add a prefix, for example 900, resulting in the dialed string 90012345.

In order to bill such calls using PortaBilling, you must create a VOIP to Vendor Connection with remote IP=MVTS-UA and a vendor tariff of your choice.

There are two steps to configuring this prefix:

1. PortaBilling configuration file [~porta-billing/etc/porta-billing.conf]
[MVTS}
MVTSLocalNumberPrefix=900

2. MVTS dialplan file [usr/local/mvts/cfg/dialpeer.cfg]

This is required in order to create the following peer:

```
[DP900]
dst_pattern=900[0-9]*
gateway=ENDPOINTS
hunt_stop=1
```

Adding this dialpeer will avoid the use of external RADIUS routing for these calls. Advanced features such as UM and follow-me are not supported for such calls.

There is no need to create rates for prefix 900, since this prefix is stripped by the billing engine.

Testing the MVTS configuration

1. Change dir to the MVTS base directory:

```
cd /usr/local/MVTS
```

2. Run MVTS with a command:

```
bin/mp_kernel.d.x -c -x cfg/meraproxy.cfg
```

3. Connect to the Mera console:

```
bin/mp_shell.sh
```

4. Make sure your dialpeer is configured properly; type in:

```
show dial 12345
```

The output should look like this:

```
Search routing for
dst_number      :12345
```

```
src_number      :
group           :DEFAULT
> START NEW SEARCH
> checking dialpeer DP0 ... valid
> translating aliases and group
> dst_number    :12345
> src_number    :
> dstbillnumber:12345
> srcbillnumber:
> group         :DEFAULT
> try to find destination gateways ...
> EXTERNAL reference (Ask remote system for routing)
SUCCESS
destination endpoint: id =EXTERNAL, new aliases
after route search
dst_number      :12345
src_number      :
src_billnumber:
dst_billnumber:12345
group           :DEFAULT

----- TOTAL 1 ROUTE PATH
```

Adv. Cases:
Routing to the remote GK
Special settings for one of the vendors
Auth. Remote gw by IP address
Auth by tech prefix

3. PortaBilling Configuration

PortaBilling100 Configuration

Please see the [PortaBilling Administrator Interface](#) for detailed instructions on how to navigate in and utilize the web interface, as well as detailed explanations of particular fields.

Checklist

Print the following page and use it to check off the operations you have completed when performing system setup according to the instructions in this chapter. Please be sure to perform all of the operations (i.e. all of the boxes should be checked), otherwise the service will not work.

Operation	Done
General configuration	
Fill in company data in Company Info	[]
Specify base currency	[]
For any other currency you plan to use, specify the exchange rate source and define exchange rates	[]
Create all the required destinations for off-net calls	[]
Network configuration	
Create a node for MVTS	[]
Rating configuration	
Create a tariff A, which will apply to your customers (people who send traffic to you)	[]
Enter rates in tariff A for the destinations your customers want to call	[]
Create a tariff B, which describes your termination costs and routing for off-net calls (make sure it has a Routing type!)	[]
Enter rates in tariff B for the destinations you plan to call	[]
Create a "Traffic exchange" product	[]
Create one accessibility entry for this product, using the MVTS node and tariff A	[]
Create an off-net calls vendor	[]
Create a connection for this vendor using tariff B	[]
Account provisioning	
Create a retail customer who will use the "Traffic exchange" service	[]
Create several accounts for this customer, with an account ID identical to the remote gateway IP address	[]
Testing	
Make a test call	[]

Initial PortaBilling configuration

The following steps are normally performed only once, after the system has been installed:


- Visit **Company Info** from the main menu. Enter information about your company and set up a base currency. Of course, this does not limit your operations to this currency only. However, different currencies will be converted to the one you specify here on reports such as cost/revenue.
- From the main menu, choose **Users** and create login entries for users who will be working with the system. It is not recommended that the default PortaBilling root user (**pb-root**) be used for any operations other than initial setup.
- Make sure you are able to login as the newly-created user, and change the password for the **pb-root** user.

NOTE: It is possible that you will require assistance from PortaBilling support staff in the future. In order to provide support, they will need access to the web interface. Therefore, when submitting a problem report please either provide them with the new password for the **pb-root** user, or create a special user for them.

- If you plan to do billing in more than one currency, define these in **Currencies** and specify the exchange rates in **Exchange Rates**.

Create destinations

This step is only required if you have not previously defined the necessary destinations. There are two ways of entering new destinations into the system:



- One-by-one, using the  **Add** functionality on the web interface
- By bulk update, uploading the destinations from a file

NOTE: A file with the default destination set is supplied with PortaBilling. You can download it and then upload it to the server. However, it is possible that your business will require different types of prefixes, so please examine the data in the file before uploading.


Creating destinations one-by-one

1. In the Management section of Admin-Index, choose **Destinations**.



2. Click the  **Add** button.
3. Fill in the required information. This includes the phone prefix and country. The country subdivision is optional. You can use the **Description** column to store additional information about the destination (for example, if it is a mobile or fixed number).
4. Click  **Save**.
5. Repeat steps 2-4 for each additional destination.

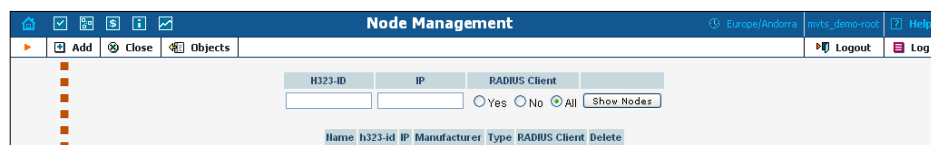
Uploading a set of destinations from a file

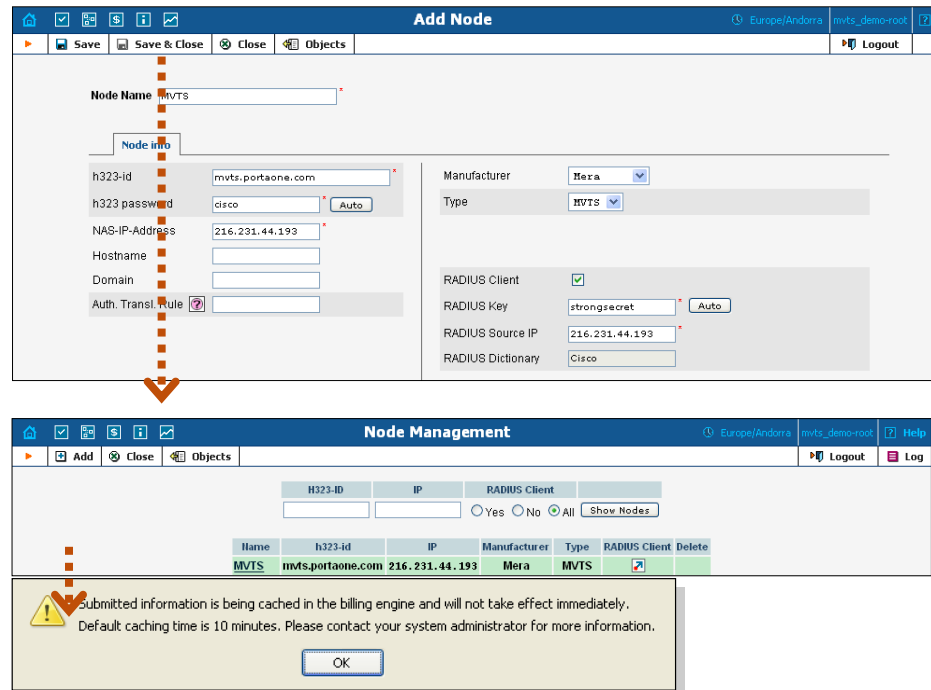
1. In the Management section of Admin-Index, choose **Destinations**.
2. Click on **Get default set** to download a set of destinations as a CSV (Comma-Separated Values) file.
3. Open this file in Microsoft Excel or any other suitable program. Edit the data if required.
4. Save the file and close it.
5. Switch back to the PortaBilling web interface, and click **Upload** on the Destinations screen.
6. Type in the filename for the file you have edited, or click on the **Browse...** button and select the file.
7. Click  **Save & Close**.



Create nodes

You must create at least one node to represent your MVTS server. If other gateways are part of your network (Cisco, Quintum), you must enter these as well.

NOTE: Only your own gateways need to be entered as nodes. Remote gateways belonging to a partner/carrier, or those legally belonging to you but used solely by your customer(s), are not considered nodes.





1. In the Management section of Admin-Index, choose **Nodes**.
2. In the Node Management window, click the  **Add** icon.
3. Fill in the New Node form:
 - **Node name** – A short descriptive name for this node (this will be used in the select menus).
 - **H323-ID** – h323-id (recommended: `hostname.domainname`)
 - **H323 Password** – If you plan to send calls from MVTS to your Cisco gateways, where the default Cisco remote IP authentication script is used, enter `cisco` here.
 - **NAS-IP-Address** – IP address of the MVTS server.
 - **Auth. Translation rule** – Used to convert a dialed number to the desired format.
 - **Manufacturer** – Select **Mera**.
 - **Type** – VoIP node type; select **MVTS**.
 - **Radius Client** – Make sure this box is checked.
 - **Radius Key** – Create a value (using the general rules for password creation). Make sure the same value is entered in the `meraproxy.cfg` file.
 - **Radius Source IP** – IP address of the MVTS server.
4. Click  **Save&Close**.
5. Repeat steps 2-4 until all of your nodes have been entered.


NOTE: There is some propagation delay between the database and the Radius server configuration file, but no more than 15 minutes.

Create a tariff


A tariff is a single price list for call services. A tariff combines:

- conditions applicable to every call regardless of the called destination
- per destination rates.

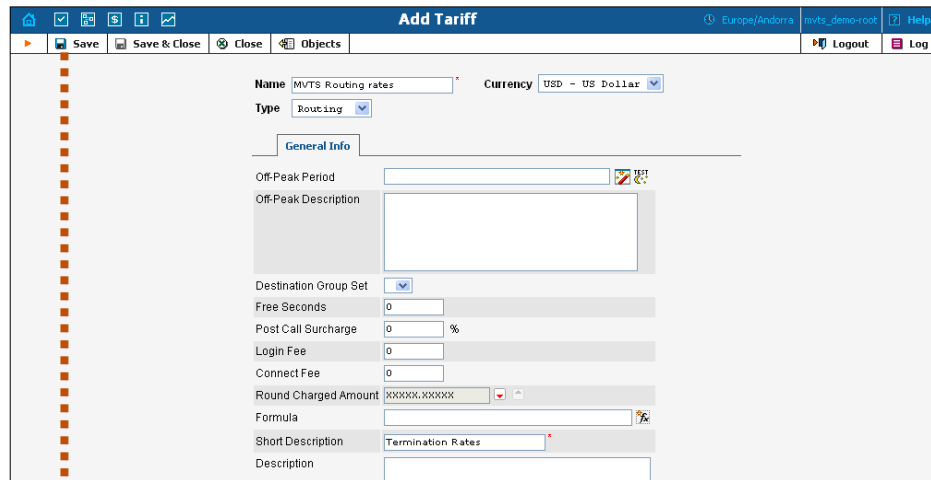
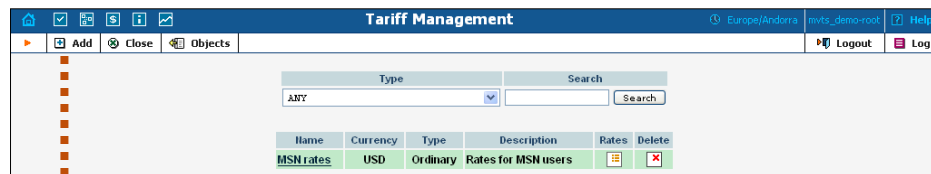
Normally, you will need a separate tariff for each of your customers.

1. In the Management section of Admin-Index, choose **Tariffs**.
2. On the Tariff Management page, choose  **Add**.
3. Fill in the **New Tariff** form:
 - **Name** – A short name for the tariff object. This is the name you will see later in the select menus.
 - **Currency** – Indicates the currency in which pricing information is defined. All pricing information for a single tariff must be defined in the same currency.

NOTE: The currency for a tariff is chosen only once, and cannot be changed later.

- **Type** – Choose “Ordinary” for a tariff for your customers, and “Routing” for your vendors.
- **Off-peak Period** – Defines the off-peak period. Click on the off-peak period wizard icon () to summon the wizard, which will help you construct a correct period definition. Click **Help** to get more information on period format definition. If you do not differentiate between peak and off-peak rates, just leave this field blank.
- **Off-Peak Description** – A description of the off-peak period, automatically filled in by the off-peak period wizard. You do not need to fill in this field.
- **Destination group set** – If you wish to enter rates in the tariff for a whole group of prefixes at once, rather than for each individual prefix, you should create a destination group set and destination groups beforehand. Leave this select menu empty for now.
- **Free seconds** – The number of free seconds allowed for each call. In order to use free seconds, the call length must be at least one billing unit (i.e. the first interval; see the ‘Enter Rates’ section above).
- **Post Call Surcharge** – A percentage of the charged call amount.
- **Login Fee** – The amount to be charged immediately after initial user authentication (i.e. after the user enters his PIN).
- **Connect Fee** – The amount to be charged for each connected call (of a non-zero duration).

- **Round charged amount** – Instead of calculating CDRs with 5-decimal-place precision, round up CDR amount values (e.g. to cents, so that 1.16730 becomes 1.17).
 - **Formula** – The default rating formula which will be applied to each rate created in the tariff. If you leave this empty, “old-style” rating will be used.
 - **Short Description** – A short tariff description. This will be shown in the rate lookup on the admin interface and the self-care pages for your accounts and customers. For example, the short description for the cust-ABC-Easy Call-1800 tariff will provide better information for your reseller ABC, who will be using this tariff, e.g. “EasyCall – via a toll-free number”. This field is mandatory; if you are unsure what to enter here, enter the same name as for the tariff.
 - **Description** – An extended tariff description.
4. Click **Save**.



Destination	Country	Routing	Interval, sec.	Price, USD min	Effective From						
Description	Route Category	Preference	Huntstop	First	Next	First	Next	First	Next	YYYY-MM-DD	HH24:MI:SS
1		Default	5		1	1	0.015	0.015		Immediately	


Destination	Country	Routing	Interval, sec.	Price, USD min	Effective From						
Description	Route Category	Preference	Huntstop	First	Next	First	Next	First	Next	YYYY-MM-DD	HH24:MI:SS
1	UNITED STATES US prefix	Default	5		1	1	0.01500	0.01500		2005-11-22 14:31:52	


Enter rates


Rates are per-destination prices. Please refer to the [System Concepts](#) chapter for more details on billing parameters.

Managing rates online

Managing rates online is very convenient for maintaining existing rate tables, as well as for reference purposes. In the case of new price lists or major updates, the offline method is better.

1. On the Tariff Management page, you will see a list of available tariffs. Click the  **Rates** icon next to the name of the tariff. When you are



in Tariff Management for a particular tariff, click on  **Rates** in the toolbar.

2. In the **Edit Rates** screen, click  **Add**.
3. Fill in the required information:
 - **Destination** – A destination prefix may be entered directly, e.g. **47** for Norway, or you can access the destinations directory by clicking the **Destination** link (in the column header). Here you can find the desired prefix by country name.

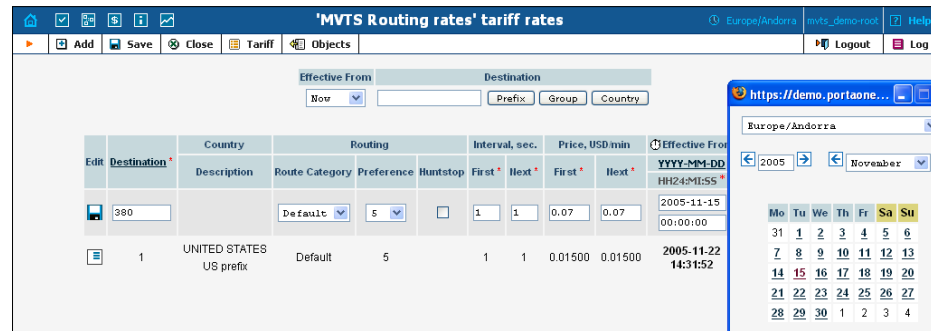
NOTE: The phone prefix you are trying to create a rate for must already exist in Destinations.



- **Interval First** – first billing unit in seconds
- **Interval Next** – next billing unit in seconds
- **Price First** – per-minute price for first interval
- **Price Next** – per-minute price for next interval
- **Off-peak Interval First**– first billing unit in seconds for off-peak time
- **Off-peak Interval Next** – next billing unit in seconds for off-peak time
- **Off-peak Price First** – first interval per-minute price for off-peak time
- **Off-peak Price Next** – next interval per-minute price for off-peak time

NOTE: Off-peak fields appear only if an **off-peak period** has been defined for the tariff.

- **Formula**  – Launches the wizard for creating a custom rating formula.
- **Effective from** – If you want this rate to take effect sometime in the future, you can either enter a date manually, or use the calendar (click on the DD-MM-YYYY link). Click on the  Stop Watch icon to make the rate effective **immediately**.

NOTE: When using the calendar, you may specify a different time zone than the current one for the date you are entering. PortaBilling will then automatically adjust the time.



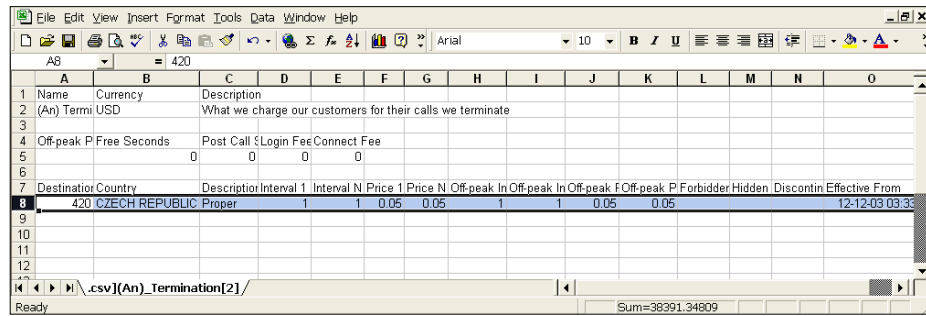
- The **Hidden**, **Forbidden** and **Discontinued** flags are optional.
4. Click the  **Save** button in the toolbar, or the  icon at the left end of the row.
 5. Repeat the preceding steps if you need to enter more rates.

Managing rates offline

NOTE: Templates are available in PortaBilling, a powerful tool for uploading rates from custom format data files. However, in this particular example we assume that you are preparing data in the default PortaBilling format.

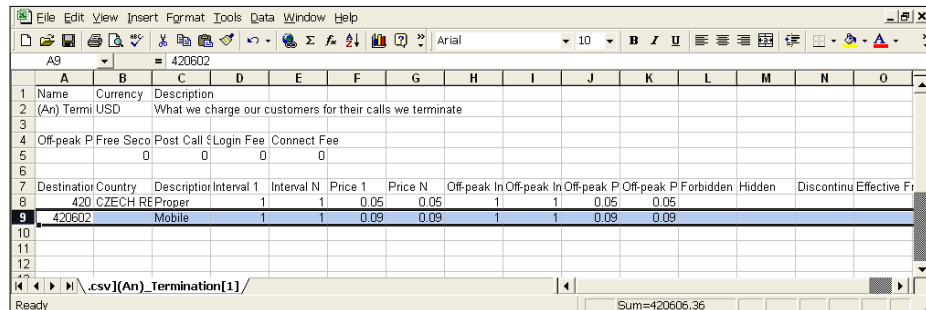
The rates table may be prepared using a spreadsheet processor (i.e. Microsoft Excel) and then easily imported into PortaBilling. This is very convenient should you wish to make many changes. For example, you could increase all prices by 10%.

1. If you are not in Tariff Management for your tariff, go to the main menu, click on **Tariffs**, and then click on the tariff name.
2. In the Edit Tariff window, move the mouse pointer over the **Download** button and hold it there until a popup menu appears. Choose the **Now** menu item and click on it. This will download the current set of rates (empty), and will also provide you with an overview of the file structure.
3. You will see the **File Download** dialogue, and will be prompted to save the file or open it from its current location. We recommend that you first save the file to the folder you will be using to store tariff data files in the future, and then open it in Excel.
4. You should see something similar to the screenshot below:




1	Name	Currency	Description															
2	(An) Termi	USD	What we charge our customers for their calls we terminate															
4	Off-peak P	Free	Seco	Post Call	Login Fee	Connect Fee												
5		0	0	0	0	0												
7	Destination	Country	Description	Interval 1	Interval N	Price 1	Price N	Off-peak	In Off-peak	In Off-peak	P Off-peak	P Off-peak	P Forbidden	Hidden	Discontin	Effective	From	
8	420	CZECH REPUBLIC	Proper	1	1	0.05	0.05	1	1	0.05	0.05						12-12-03	03:35

5. Edit the file by adding more rows with rate data, so that it looks similar to the screenshot below.
6. Note that the **Country** and **Description** columns are only for reference purposes, and are ignored during import. Also, when using the default template you must fill in the Off-peak columns even if your tariff does not have an off-peak period (use the clipboard to easily copy values to the four peak columns).
7. Also note that you may only use those phone prefixes which you have already defined as destinations (see **Create destinations** above).



1	Name	Currency	Description															
2	(An) Termi	USD	What we charge our customers for their calls we terminate															
4	Off-peak P	Free	Seco	Post Call	Login Fee	Connect Fee												
5		0	0	0	0	0												
7	Destination	Country	Description	Interval 1	Interval N	Price 1	Price N	Off-peak	In Off-peak	In Off-peak	P Off-peak	P Off-peak	P Forbidden	Hidden	Discontin	Effective	From	
8	420	CZECH REPUBLIC	Proper	1	1	0.05	0.05	1	1	0.05	0.05							
9	420602		Mobile	1	1	0.09	0.09	1	1	0.09	0.09							



8. Save the file in Excel. You will probably get a warning from Excel that your file “*may contain features that are not compatible with CSV (Comma delimited)*”. Ignore this, and choose **Yes** to retain CSV format.
9. Close the file in Excel. If you performed step 6, then disregard the message “*Do you want to save the changes you made?*”, as this only results from the fact that your format is not the native Excel XLS format.
10. Go back to the PortaBilling web interface and the **Edit Tariff** screen.
11. Click on the **Upload** button.
12. Either enter the name of your file manually, or click **Browse...** and choose the file.
13. Click  **Save & Close**. You should return to the **Edit Tariff** screen, where a message will inform you of the status of the import. Also, you will receive email confirmation of the tariff upload. If any operation has failed, you will receive the portion of the data which was not uploaded as an attachment, so that you can try to import it later.

You can verify your work using the **Edit Rates** feature. After you have done this, go to the **Main** menu (by clicking on the **Home** icon).

Create all the required tariffs





Repeat the **Create Tariff** and **Enter Rates** steps until you have created the following:

- A tariff for each type of service you provide to your customers. For instance, if you plan to provide a wholesale termination service as well as end-user VoIP calls, you will need two separate tariffs.
- A tariff with termination costs for each termination partner you have.
- If you have resellers, also create the tariffs you will use for charging each of them. Do not create tariffs which will apply to your resellers' subscribers yet. First create your customers, and then return to this step.

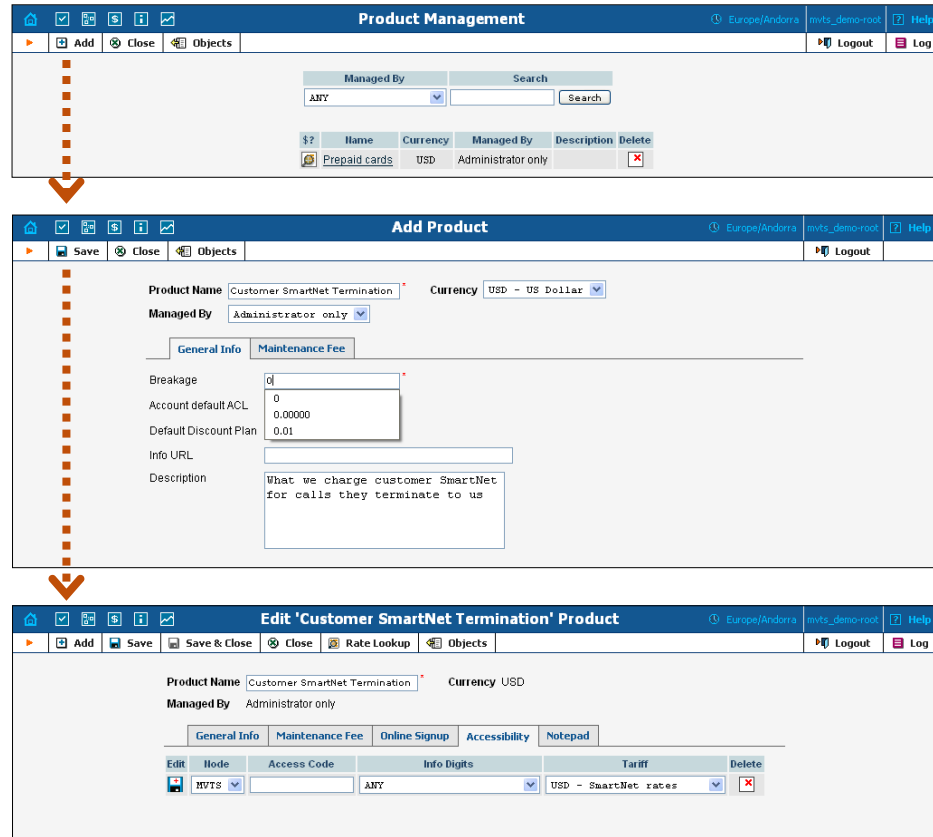
Create a product

Each of the remote customer gateways will be represented as an account, and billed accordingly. Hence we need to create a product for the account in order to have a defined way of billing it. If you have specific per-customer rates/tariffs, then you will need a product for each customer.


1. In the Management section of the Admin-Index page, choose **Products**.
2. On the Product Management page, click the  **Add** icon.
3. Fill in “Add product” under the General Info tab:
 - **Product name** – Product object name.
 - **Currency** – Product currency. Only tariffs which have the same currency are permitted in the accessibility list.
 - **Managed by** – The default is “Administrator”.
 - **Breakage** – A leftover balance which is considered “useless” (for statistical purposes). Accounts with a balance below breakage are counted as *depleted*. This does not affect account authentication or authorization, so the account can still make calls if there is enough money left to cover at least the first interval.
 - **Account default ACL** – The access level assigned by default to new accounts created with this product. The ACL determines which operations may be performed by accounts on the self-care pages. The default is “Account self-care” (pre-defined ACL), which allows all possible operations.
 - **Default discount plan** – The default value is “None”.

- **Info URL** – A URL where current product info is displayed.
 - **Description** – A description of the intended use of this product.
4. Click  **Save**.

Click on the **Accessibility** tab to edit this product’s accessibility.



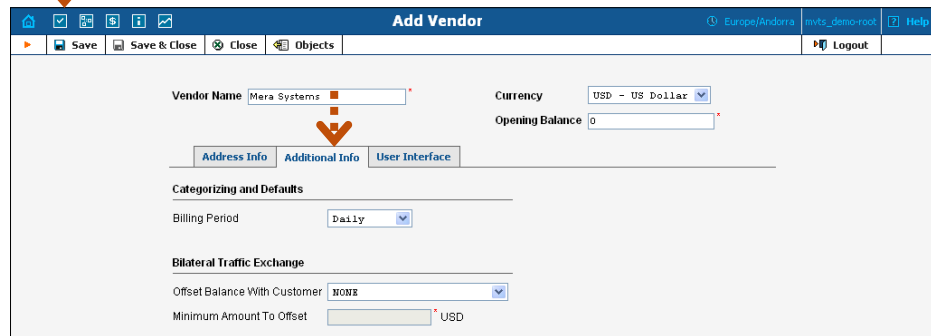
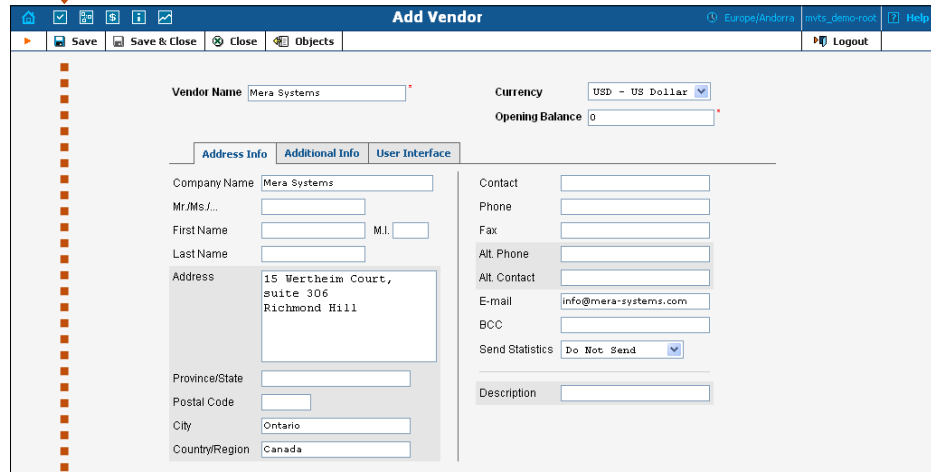
Enter the node and tariff into the product’s accessibility list

1. After selecting the Accessibility tab, click on the  **Add** icon.
2. Choose **MVTS** as the node, select the tariff with which you want to bill your customer, and leave the CLD field empty.
3. Click **Save** to save this accessibility entry.

Create vendors

This step is only required if you have not entered information about your vendors into the system before. Vendors are your termination partners or the providers of incoming toll-free lines.

1. In the Management section of the Admin interface, choose **Vendors**.
2. On the Vendor Management page, choose **Add**.



- Fill in the **New Vendor** form. Please note that there are two tabs available on the screen. The most important fields are:

Main form (top)


- **Vendor name** – Short name for the vendor object; this will be used on the web interface.
- **Currency** – The currency in which this vendor charges you.
- **Opening balance** – Starting balance for the vendor; the default is zero.

Additional info

- **Billing period** – Split period for vendor statistics.

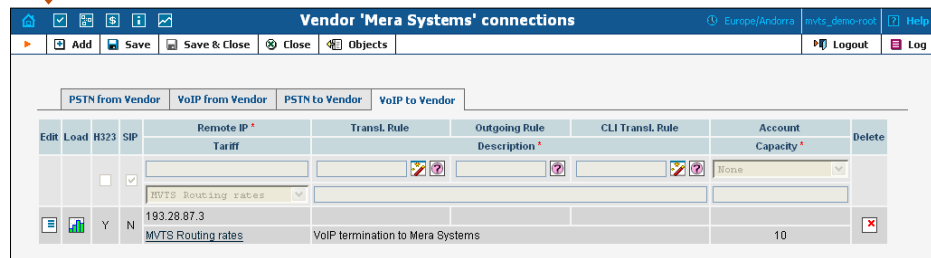
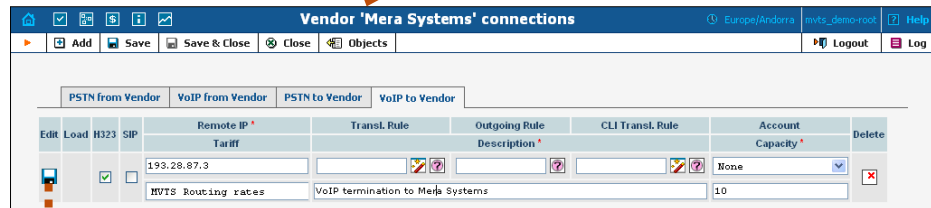
User-Interface

- **Time zone** – The time zone which the vendor uses for his billing period. Statistics will be divided into periods according to this time zone.

- Click  **Save & Close**.
- Repeat steps 2-4 to add all of your vendors.

Define connections

- In the Management section of the Admin interface, choose **Vendors**.
- Click on the **Connections** icon next to the vendor name.

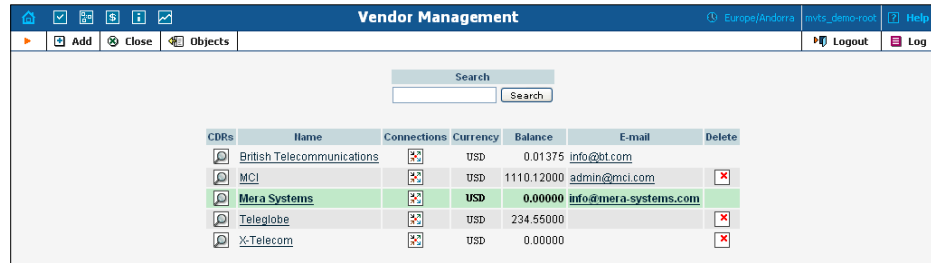


3. Choose the connection type (e.g. **VoIP to Vendor**) by clicking on the corresponding tab.
4. Press **Add** to add a new connection.
5. Fill in the connection information. Enter the remote gateway's IP address. Choose the tariff which defines your termination costs for this connection/vendor. **Description** and **Capacity** are mandatory for all connection types.
6. If the vendor requires a phone number format different from the one you use (e.g. the E.164 number 4202111222 is sent to the vendor as 34562#4202111222) you should set up translation rules to properly handle this situation. Use **Translation Rule** to convert the phone number from the vendor's format to the one you use (i.e. 34562#4202111222 to 4202111222).
7. Click **Save**.
8. Repeat steps 3-5 to add more connections to the same vendor, then click **Close** to exit to the **Vendor Management** screen.
9. Repeat steps 2-7 to add connections for other vendors.

Create all vendors

Since you are setting up a wholesale traffic exchange platform, you will probably have more than one vendor. Create all of your vendors

according to the instructions given above. Every vendor will have his own termination tariff and set of connections.

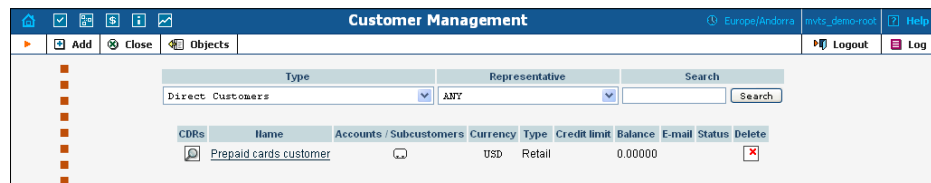


CDRs	Name	Connections	Currency	Balance	E-mail	Delete
	British Telecommunications		USD	0.01375	info@bt.com	
	MCI		USD	1110.12000	admin@mci.com	
	Mera Systems		USD	0.00000	info@mera-systems.com	
	Teleglobe		USD	234.55000		
	X-Telecom		USD	0.00000		

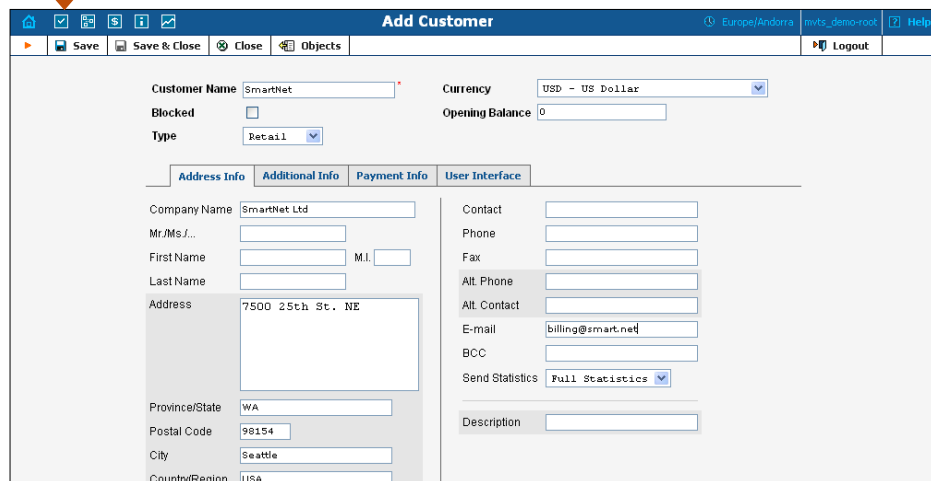
Create a customer

A customer is an owner of accounts. The customer’s contact information is used to distribute generated account data and account usage information.

1. In the Management section of Admin-Index, choose **Customers**.
2. On the Customer Management page, choose **Add**.
3. Fill in the **New Customer** form. Please note that there are several tabs with extra information available on the screen. The most important fields are:



CDRs	Name	Accounts / Subcustomers	Currency	Type	Credit limit	Balance	E-mail	Status	Delete
	Prepaid cards customer		USD	Retail		0.00000			



Customer Name SmartNet **Currency** USD - US Dollar

Blocked **Opening Balance** 0

Type Retail

Address Info | **Additional Info** | **Payment Info** | **User Interface**

Company Name: SmartNet Ltd

Mr./Ms./...:

First Name: M.I.

Last Name:

Address: 7500 25th St. NE

Province/State: WA

Postal Code: 98154

City: Seattle

Country/Region: USA

Contact:

Phone:

Fax:

All Phone:

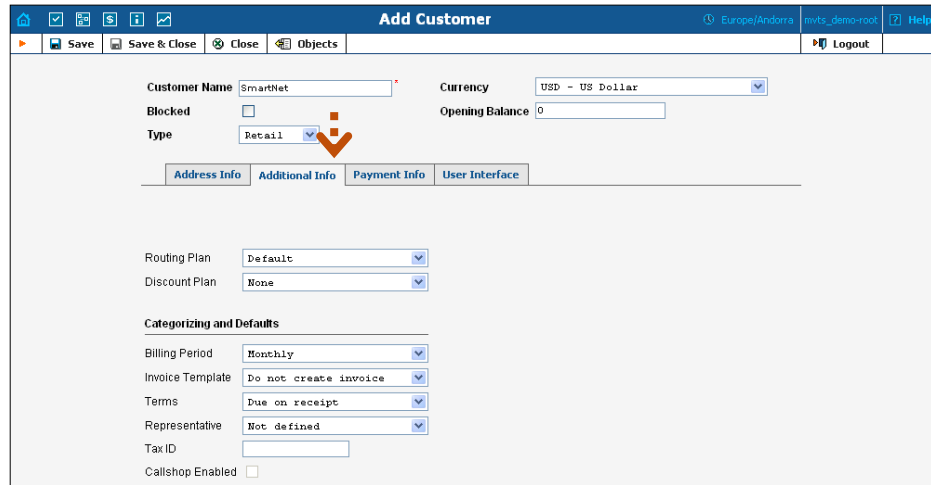
Alt. Contact:

E-mail: billing@smart.net

BCC:

Send Statistics: Full Statistics

Description:



Add Customer Europe/Andorra mvts_demo-root Help

Save Save & Close Close Objects Logout

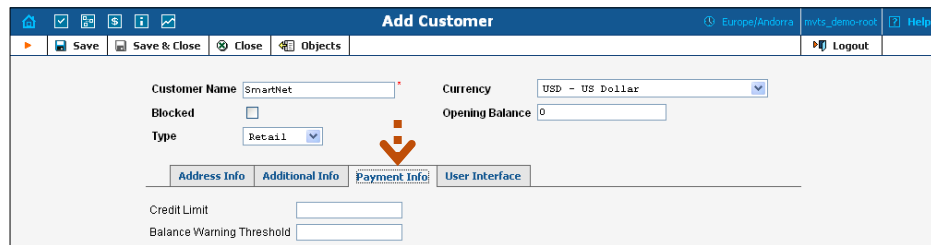
Customer Name SmartNet Currency USD - US Dollar
 Blocked Opening Balance 0
 Type Retail

Address Info Additional Info Payment Info User Interface

Routing Plan Default
 Discount Plan None

Categorizing and Defaults

Billing Period Monthly
 Invoice Template Do not create invoice
 Terms Due on receipt
 Representative Not defined
 Tax ID
 Callshop Enabled



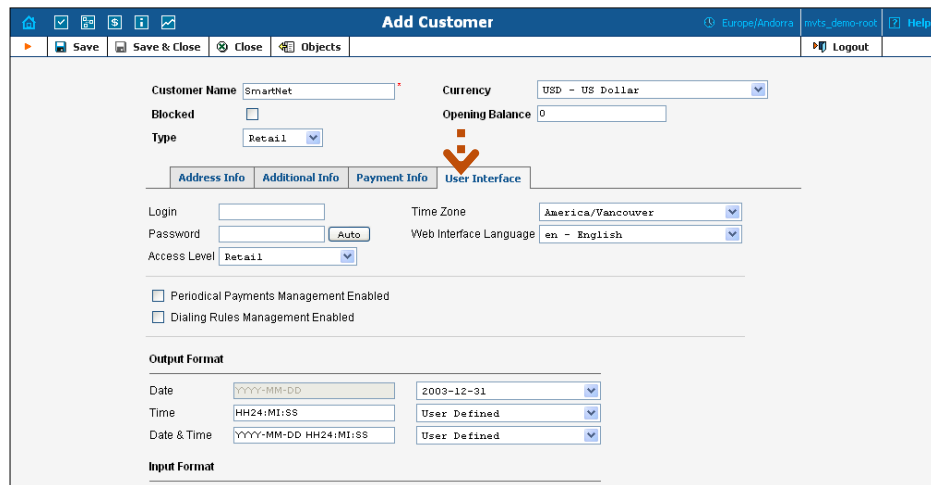
Add Customer Europe/Andorra mvts_demo-root Help

Save Save & Close Close Objects Logout

Customer Name SmartNet Currency USD - US Dollar
 Blocked Opening Balance 0
 Type Retail

Address Info Additional Info Payment Info User Interface

Credit Limit
 Balance Warning Threshold



Add Customer Europe/Andorra mvts_demo-root Help

Save Save & Close Close Objects Logout

Customer Name SmartNet Currency USD - US Dollar
 Blocked Opening Balance 0
 Type Retail

Address Info Additional Info Payment Info User Interface

Login Password Auto Time Zone America/Vancouver
 Access Level Retail Web Interface Language en - English

Periodical Payments Management Enabled
 Dialing Rules Management Enabled

Output Format

Date YYYY-MM-DD 2009-12-31
 Time HH24:MI:SS User Defined
 Date & Time YYYY-MM-DD HH24:MI:SS User Defined

Input Format

Main form (top)

- **Name** – Short name for the customer object; this will be used on the web interface.
- **Currency** – The currency in which this customer is to be billed.
- **Opening balance** – Starting balance for the customer; the default is zero.

- **Type** – Define whether this is a reseller or a retail (direct) customer. (Normally, most of your customers will be retail customers. Only if a customer resells your services, while you provide services and billing to his subscribers, will he be created as a reseller.)

Address info tab

- **Email** – Email address for distribution of accounting information. When the billing period is over, a list of CDRs and other statistics will be sent to this address.
- **Bcc** – Blind carbon copy in an email; this may be used for debug and archiving purposes.
- **Send Statistics–summary only** – Distribute a summary only, and do not attach a details file; this might be useful in the case of a large amount of calls. Other options are **full statistics** or **do not send**.

Additional info tab

- **Billing period** – The frequency of accounting information distribution. The available billing periods are:
 - **Daily** – One day, from midnight to midnight; sent on the next day.
 - **Weekly** – [Mon-Sun] inclusive; sent on Monday.
 - **Bi-weekly** – [1-15] inclusive, sent on the 16th day; and [16-last day] inclusive, sent on the 1st day of the next month.
 - **Monthly** – [1-last day] inclusive; sent on the 1st day of the next month.


Payment info tab

- **Credit limit** – If left empty, there is no credit limit for this customer.
- **Balance warning threshold** – The customer can be notified by email when his balance is dangerously close to the credit limit and his service will soon be blocked. Here you can enter the value for the warning threshold, as follows:
 - As a percentage (e.g. 90%). A warning will be sent when the customer's balance exceeds this percentage of his credit limit. Thus, if his credit limit is \$1000.00 and the threshold is 90%, a warning will be sent as soon as the balance is over \$900.00. This is only applicable when the customer has a positive credit limit.

- As an absolute value. A warning will be sent as soon as the balance exceeds the specified value.


User Interface

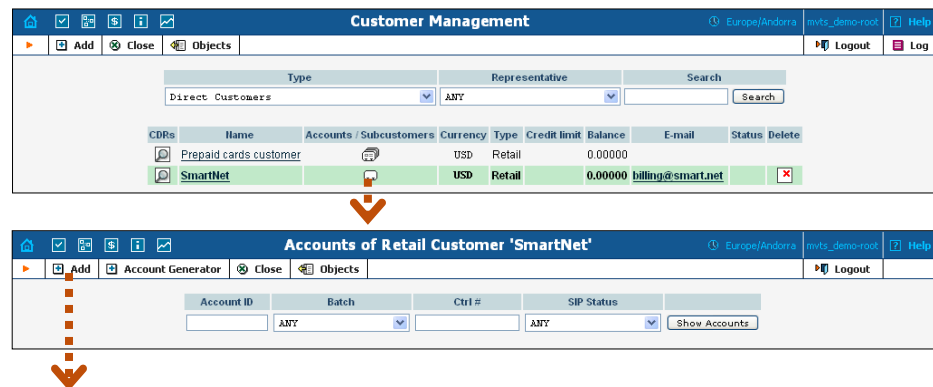
- **Time zone** – This parameter serves two purposes. First of all, it defines in which time zone the customer will see his CDRs. It also defines which time zone will be used to divide the customer’s billing periods. For example, if you choose America/New York with a monthly billing period here, the customer’s invoice will cover the period starting at 00:00 EST on the 1st of every month.
- **Web interface language** – The language to be used on the customer’s self-care web interface.

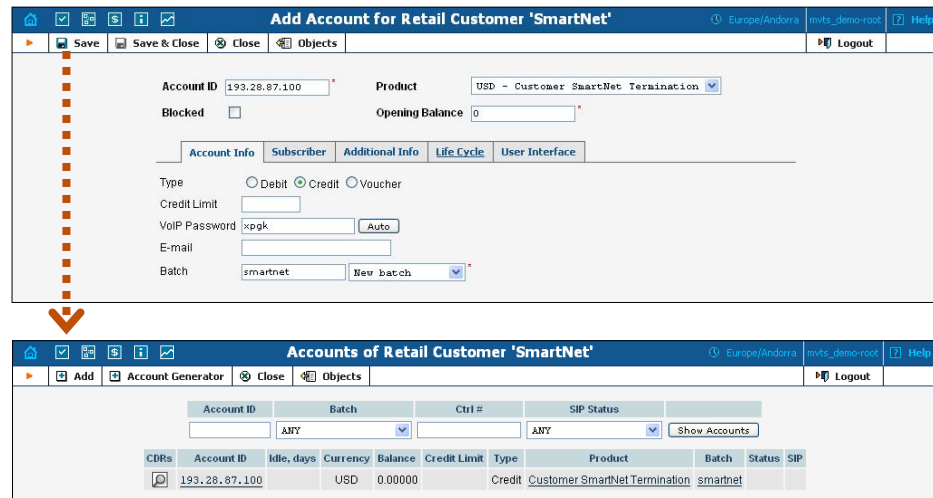
Click  **Save & Close** to save your work. See the [PortaBilling Administrator Interface](#) for more information.


Create Accounts

NOTE: Before generating accounts for a production system, read the section on “Provisioning prepaid accounts”.

1. Go to the **Customers** screen (the screen which contains a list of customers). It should look like the screenshot below.
2. Next to the customer name, click on the  icon (in the **Accounts** column), which will take you to the account management for that customer.





3. Now click on  **Add**.
4. Fill in the “Add account” form:
 - **Account ID** – Identification of the account (value to be sent in the **User-Name** attribute). For an account representing a remote gateway, this is normally an IP address.
 - **Product** – Choose the product which you would like your accounts to use.
 - **Blocked** – Check this if you want to create the account as initially blocked.
 - **Opening balance** – The initial balance on the card. For credit accounts, the opening balance is normally zero.

Account info tab

- **Account type** – Account type; select “credit”.
- **Credit limit** – Maximum allowed credit.
- **VoIP password** – Password for authentication/authorization. If you are using the default Cisco remote_ip_authenticate script, enter **cisco** here.
- **Batch** – A batch is a management unit for accounts. A batch name is alphanumeric. You can type a new name here, or use the existing name in order to generate more accounts for the same batch.

Additional Info tab


- **Preferred language** – This is a custom attribute which is transferred to the IVR. Leave English here if you are not sure whether your IVR supports this feature.
- **Redirect number** – Redirect number (discussed in the **Advanced features** section); leave this empty.

- **UM enabled** – Check to enable.
- **IP phone** – Default = None.
- **IP phone port** – Default = None.
- **Follow me** – Check to enable.
- **E-commerce** – Check to enable.
- **Discount plan** – Enter the product’s default or choose None.
- **Music on hold** – Enter the customer’s default or choose None.

Life Cycle tab

- **Activation date** – Account activation date.
- **Expiration date** – Account expiration date.
- **Lifetime** – Relative expiration date; the account will expire on “first usage date” + “lifetime” days. If you do not want to use this feature, leave the field blank.

User Interface tab

- **Login** – The login name this account will use to access the self-care pages. This can be the same as the account ID, or a different name may be chosen for increased security. This field is mandatory.
 - **Password** – The password for the self-care pages.
 - **Time zone** – When an account owner accesses the web self-care pages to see a list of his calls, the time will be shown in the time zone most appropriate for him.
 - **Web interface language** – The language to be used on the customer’s self-care web interface.
5. Click  **Save & Close**; a confirmation screen will indicate that the account has been created.
 6. Repeat steps 3-5 if the customer has more than one remote gateway.

Call Routing

When an entity on your network (for instance, a gateway or a SIP server) establishes an outgoing call, it must determine where the call is being sent to. To do this, it can use its internal configuration (for example, a dial-peer on the gateway) or an external entity (e.g. a gatekeeper), or it may ask billing for a list of possible routes. This last method has several advantages: the routing configuration is in one central location, and billing can use information about termination costs to choose the best route (least-cost routing).

When a call goes through the MVTS server, MVTS may:

- Direct the call to one of the registered end-points, if the called number belongs to the registered agent.
- Route the call to one of the gateways for termination, according to the routing rules specified in PortaBilling.

Routing SIP/H323 on-net calls

PortaBilling can inform MVTS whether the dialed number is actually provisioned as an account, so then a call to it will be an on-net call. The MVTS server automatically maintains information about all currently registered endpoints, so it is able to determine which IP address and port the call should be routed to.

Routing off-net calls

You can use different vendors to terminate off-net calls. For example, you can terminate calls to the US either to AT&T, via a T1 connected to your gateway in New York, or to a remote gateway from Qwest.

Rate routing parameters

Ordinarily, tariffs define the termination costs for each connection to a vendor. If you create a tariff of the **Routing** type, a few more fields will be added to rates in that tariff:

- **Route category** – You can divide this into categories such as “Premium”, “Cheap” and the like, and use these categories in routing plans.
- **Preference** – Routing priority (0-10); higher values mean a higher priority, 0 means do not use this rate for routing at all.
- **Huntstop** – Indicates that no routes with a lower preference should be considered.

This allows you to easily manage both termination costs and routing from a single location on the web interface. Thus, when such a routing tariff is associated with a connection, you can send calls for termination to all prefixes for which rates exist in the tariff.

Multiple routes

It is dangerous to have only one termination partner: if their server is down, your customers will not be able to make any calls. Normally, you will try to find several vendors and enter their rates into the system. Each connection to a vendor (with a routing tariff) will produce one possible route, and PortaBilling will arrange these according to cost or your other preferences, thus providing fail-over routing.

Routing plans

Routing preferences in the rate allow you to specify that, for example, you would rather send a call to MCI than to T-Systems. However, this decision is “global”, and so will apply to all calls made in your system. But what if you would like to use MCI first for customer A, while T-Systems should be the first route for customer B, and customer C should be routed to MCI only?

This can be accomplished using routing plans. A routing plan defines routes for which categories are available, as well as in which order they should be arranged (route categories with a higher order value are tried first). For instance, in the example above MCI may be assigned as the “Normal” route category and T-Systems as the “Premium” category. After that, three routing plans will be created:

- **Quality** – includes first the Premium and then the Normal routing categories
- **Ordinary** – includes first the Normal and then the Premium routing categories
- **Cost-efficient** – includes only the Normal routing category

Thus, depending on which routing plan is assigned to the current customer, the system will offer a different set of routes. There is one routing plan which always exists in the system – **Default**. This is assigned to your customer unless otherwise specified. This routing plan uses all the available routes regardless of their route category.

Routing algorithm

The routing principle is simple:

- MVTS asks PortaBilling for routing destinations for a given number.
- PortaBilling checks each tariff with routing extensions associated with a vendor connection for rates that match this phone number. The best-matching rate in each tariff is chosen; this rate will define the routing parameters.
- A list of possible termination addresses will be produced (this will include remote IP addresses for VoIP connections and the IP addresses of your own nodes with telephony connections).
- This list will be sorted according to routing plan, routing preference and cost; entries after the first huntstop will be ignored.
- A list of these IP addresses (with optional login and password for SIP authentication) will be returned to MVTS. (To avoid extremely long delays, only a certain number of routes from the

beginning of the list are returned; the default is 15, but this can be changed in `porta-billing.conf`).

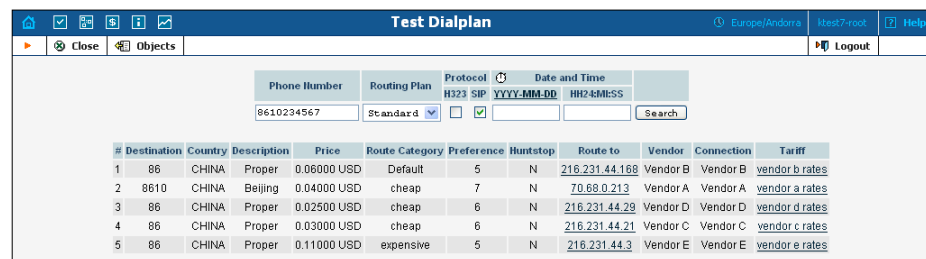
Route sorting

How exactly does PortaBilling100 arrange multiple available routes?

1. By route category. Only route categories which are included in the routing plan will be used, following the order given in the routing plan.
2. If you have multiple route categories within the routing plan, you can either merge them into the same group by assigning them the same order value, or keep each one separate, with its own order value. Then routes within the same order group of route categories will be arranged according to preference.
3. The system can arrange routes with the same preference according to cost (a comparison is made using the **Price_Next** rate parameter), so that cheaper routes will be among the first ones, or else in random fashion.

Does PortaBilling/MVTS support LCR?

Yes, we support LCR – and much more besides. In fact, “just LCR” is the simplest type of routing PortaBilling handles. If you decide not to use routing plans (one default plan for everyone) or routing preferences (same preference for all vendors), then the routing decision will be determined solely by the vendor’s cost.



The screenshot shows the 'Test Dialplan' interface with search filters and a table of routes. The search filters include Phone Number (8610234567), Routing Plan (Standard), Protocol (H323 SIP), and Date and Time (YYYY-MM-DD HH:MM:SS). The table below lists the routes found:

#	Destination	Country	Description	Price	Route Category	Preference	Huntstop	Route to	Vendor	Connection	Tariff
1	86	CHINA	Proper	0.06000 USD	Default	5	N	216.231.44.168	Vendor B	Vendor B	vendor b rates
2	8610	CHINA	Beijing	0.04000 USD	cheap	7	N	70.68.0.213	Vendor A	Vendor A	vendor a rates
3	86	CHINA	Proper	0.02500 USD	cheap	6	N	216.231.44.29	Vendor D	Vendor D	vendor d rates
4	86	CHINA	Proper	0.03000 USD	cheap	6	N	216.231.44.21	Vendor C	Vendor C	vendor c rates
5	86	CHINA	Proper	0.11000 USD	expensive	5	N	216.231.44.3	Vendor E	Vendor E	vendor e rates

Example

If you have:

1. A “Standard” routing plan, which includes the route categories “Default” (order 70), “Cheap” (order 40) and “Expensive” (order 10).
2. Six vendors (A, B, C, D, E, F) with the following rates (prefix, route category, preference, price):
 - a. 8610, Cheap, 7, 0.04
 - b. 86, Default, 5, 0.06
 - c. 86, Cheap, 6, 0.03

- d. 86 , Cheap, 6, 0.025
- e. 86 , Expensive, 5, 0.11
- f. 8610 , Premium, 5, 0.09

then when a customer with this routing plan makes a call to **8610234567**, the system will arrange the possible routes as follows:

Vendor	Parameters	Comment
B	Default, 5, 0.06	The “Default” route category is first in the route plan.
A	Cheap, 7, 0.04	This vendor has the highest preference in the “Cheap” category.
D	Cheap, 6, 0.025	This vendor has the same preference as vendor C, but a cheaper per-minute rate.
C	Cheap, 6, 0.03	
E	Expensive, 5, 0.11	This is the only vendor in the last route category.

(Vendor F was not included in the routing, since his route category is not in the customer’s routing plan).

4. SIP-HIT Configuration

SIP-HIT & MVTs Configuration

SIP-HIT can be used as a standalone H323-SIP translator. In conjunction with MVTs, it allows all combinations of SIP and H323 traffic with routing support.

Config.cfg

Edit the `config.cfg` file (located in the sip-hit base directory – `/usr/local/sip-hit` in our example). The most important task is assigning ports in the H323 and SIP sections.

```
[NETWORK]
## MVTs address. Should be x.x.x.x or x.x.x.x:port
gateway==<ip or domain name of MVTs if present>
# gateway=@MVTs@
  natmode=0
## routing. if there is no route, default_route and local
parameters - system routing table will be used
# route[0]=net_mask local_ip
# route[1]=net_mask local_ip
## the following are equivalent
# route[n]=0.0.0.0 local_ip
# default_route=local_ip
# local=local_ip      # this parameter is only for backward
compatibility and no longer recommended
GKPort=1719
localGkPort=1721

# ...

[H323]
## Boolean variable can be 1/0 or yes/no or true/false
H245Tunneling=1
FastStart=yes
# MultiplyFastStart=false
port=1748
# IdentifyAsCisco=false ## for internal use
## Alerting timeout (in ms!)
# AlertTimeout=-1

[SIP]
port=5060
## use or not Via header in SIP requests for reply
# UseViaForReply=1
## Accept or not REGISTER message
RegisterMethod=digest
#AllowRegister=true
domain=<ip or domain name of your default sip domain>

#
## Send 180 or 183 message
# SendProgress=false
```

```
##  
# AllowRedirect[0]=net_mask  
# AllowRedirect[1]=net_mask
```

SIP-HIT can run on the same host as MVTS or on a different one. If SIP-HIT shares the same host with MVTS, make sure they use different ports for H323 traffic.

Gateway.cfg

Edit the `gateway.cfg` file (located in the MVTS base directory – `/usr/local/MVTS` in our example). Besides the default PORTA section, you need to create a separate one called SIPCONV for routing calls via the SIP protocol:

```
[PORTA]  
address=<IP address of PortaBilling100 master server>  
port=1720  
gateway_mode=3  
gateway_type=1  
[PORTA]  
# just use the portasip address  
address=10.10.10.10  
port=1720  
gateway_mode=3  
gateway_type=1  
proxy_type=1  
  
[SIPCONV]  
address=<IP address of portasip if present, otherwise any IP ie  
127.0.0.1>  
port=5060  
gateway_mode=3  
gateway_type=1  
proxy_type=1  
converter=SIPConv
```

As in a non-SIP-HIT configuration, the **address** and **port** parameters are mandatory, but can be overridden by those which PortaBilling provides during call processing. For SIP, we recommend using the default SIP port 5060 and the IP of the PortaSIP proxy, if it is part of your installation.

The converter settings are the most important. This option tells MVTS to pass routed traffic via the configured converter if this gw section is used. The value must match the corresponding converter settings in `meraproxy.cfg`.

Meraproxy.cfg

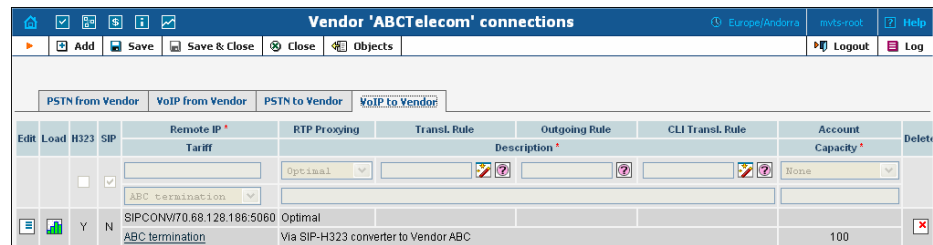
Edit the `meraproxy.cfg` file (located in the MVTS base directory – `/usr/local/MVTS` in our example). The only difference is the converter settings in the H323 section:

```
[H323]
port=1720
converter=H323Conv/<IP address SIP HIT>/1748/3/1;SIPConv/<IP
address SIP HIT>/1748/3/2
```

All converters are specified on the same configuration line and the ports must match those configured on SIP-HIT. The rest of the MVTS settings are the same as given in previous sections.

PortaBilling100 Configuration for SIP Routing via MVTS

The configuration here is exactly the same as for H323 traffic, the only difference being the mandatory gw name to be used in the VoIP to Vendor connection's remote IP settings.



Edit	Load	H323	SIP	Remote IP *	Tariff	RTP Proxying	Transl. Rule	Description *	Outgoing Rule	CLI Transl. Rule	Account	Capacity *	Delete
			<input checked="" type="checkbox"/>	SIPCONV/70.68.128.186.5060	ABC termination	Optimal		Via SIP-H323 converter to Vendor ABC			None	100	

The format is <gw-name>/<ip>[:<port>], and the gw-name must match the section given in the gateway.cfg configuration - 'SIPCONV'. The port is optional, as the converter will use the one configured in that section.

When PortaSIP is not used

When only MVTS/SIP-HIT is used, the routing decision is made on the MVTS and H323 protocol. SIP calls are converted to H323 for routing using SIP-HIT (and then converted back if routed to SIP again). From a PortaBilling100 perspective, all such connections must be marked as H323 only. This is done automatically using the following settings in the porta-admin.conf file on the PortaBilling100 slave server (/home/porta-admin/etc/porta-admin.conf):

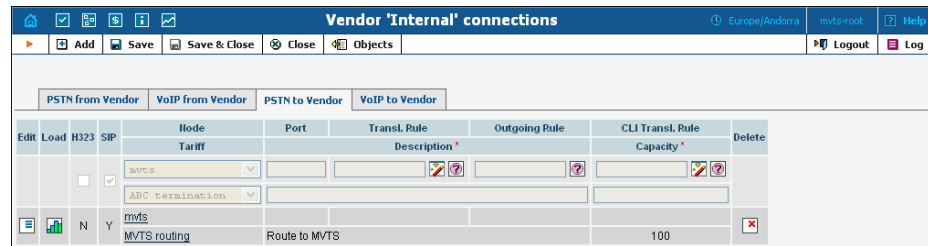
```
[AdvancedFeatures]
ShowConnectionProtocol=No
DefaultConnectionProtocol=h323
```

PortaSIP in “slave” routing mode

To utilize the full support of PortaSwitch features, including Class5 services and direct termination to registered user agents, the routing role decision must be split. When UA originates a call via PortaSIP, PortaBilling100 provides information for local termination and pure SIP-based routing. It can send the call to MVTS via a SIP-HIT converter by the “slave” route, while MVTS does the final “master” fail-over routing. In order to define separate H323 (MVTS as master) and SIP (PortaSIP as slave) routing, enable the protocol checkboxes in the `porta-admin.conf` file on the PortaBilling100 slave server (`/home/porta-admin/etc/porta-admin.conf`):

```
[AdvancedFeatures]
ShowConnectionProtocol=Yes
DefaultConnectionProtocol=h323
```

To pass all external traffic to MVTS, a special routing connection is needed. If SIP-HIT is hosted on the same IP as MVTS, configure it as a PSTN to Vendor connection using MVTS as a node:



Edit	Load	H323	SIP	Node	Port	Transl. Rule	Outgoing Rule	CLI Transl. Rule	Capacity	Delete
			<input checked="" type="checkbox"/>	mvs						
				ABC termination						
		N	Y	mvs					100	
				MVTS routing						

If SIP-HIT is on another IP, use the SIP-HIT node. Note that in this case SIP-HIT does not act as a Radius client, and authentication, authorization and accounting for all calls is performed on MVTS.


SIP is the only protocol that should be checked. The routing tariff should cover all the destinations routed on MVTS, with the routing parameters set to the default category, highest priority (10), and hunt stop set so that it is the first and only routing option on PortaSIP (so as to avoid any conflicts or routing loops). This tariff is never used to rate calls, so use a 0 price everywhere. The connection can be assigned to a special internal vendor (e.g. the same as that used for SIP-UA self-termination).

'MVTS routing' tariff rates

Europe/Andorra mvts-root Help

Add Save Close Tariff Objects Logout Log

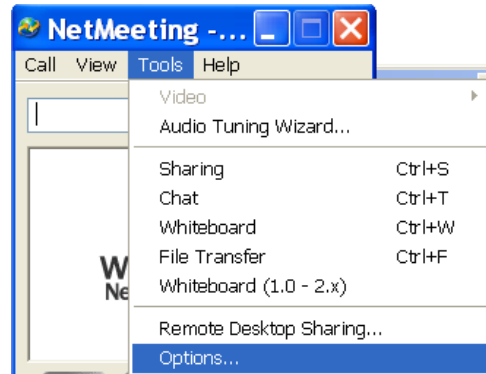
Effective From: Now Destination: Prefix Group Country

Edit	Destination *	Country	Routing			Interval, sec.		Price, USD/min		Effective From YYYY-MM-DD * HH24:MI:SS *	Delete
			Description	Route Category	Preference	Huntstop	First *	Next *	First *		
			Default	5	<input type="checkbox"/>					2006-06-30 12:28:07	
	1	Not Applicable North America	Default	10		1	1	0.00000	0.00000		

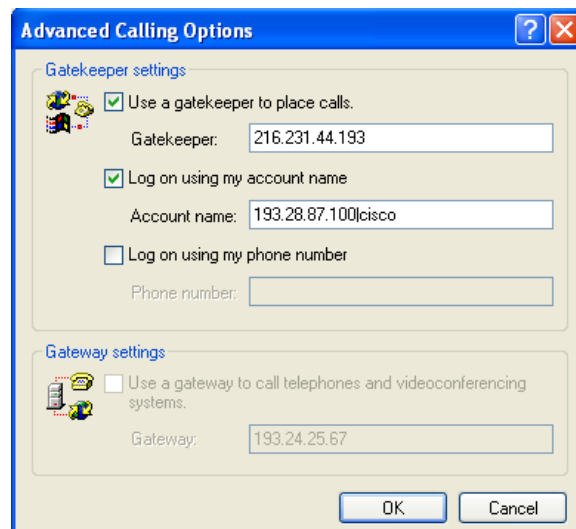
5. How to...

Configure Netmeeting to make calls via MVTS

Start NetMeeting, choose **Tools** in the main menu, then **Options**.



On the **General** tab, click the **Advanced Calling** button.



- Select the **Use a gatekeeper to place calls** checkbox.
- In the **Gatekeeper** field, enter the IP address of the MVTS server.
- Select the **Log on using my account name** checkbox.
- In the **Account name** field, enter "ID|passwd", where "ID" is your account ID, and "passwd" is the VoIP password for this account; e.g. for account 12345 with password test123 you should enter "12345|test123".
- Click OK to save your changes.

If registration to the MVTS gatekeeper is successful, you will see a blue computer icon with the popup text “Logged on to Gatekeeper” in the bottom right corner of your screen.

Terminate calls to my partner via his gatekeeper

In this case, MVTS will communicate with the remote gatekeeper to obtain the actual IP address of the remote gateway and terminate the call. Since the address of this gateway is not known in advance, you cannot enter it in the Remote IP parameter for the connection; the only address which is available is that of the remote gatekeeper.

In this case, PortaBilling/MVTS configuration should be done as follows:

1. In the MVTS configuration file `gateway.cfg`, create a new section:

```
[GKPARTNER1]
gatekeeper=GK1
```

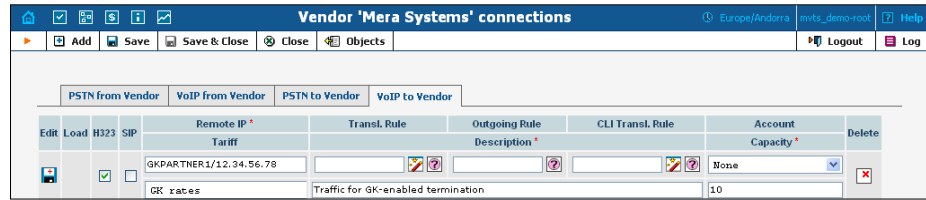
If you have more than one termination partner with their own gatekeeper, you may add more sections, e.g. [GKPARTNER2], [GKPARTNER3], and so on. You may choose any name for the section and value of the gatekeeper parameter, so long as it is unique and allowed by MVTS syntax.

2. Now you must describe every gatekeeper in the MVTS `gatekeeper.cfg` file, adding a new section there for every gatekeeper listed in the `gateway.cfg` file:

```
[GK1]
address=12.34.56.78
```

Instead of 12.34.56.78, enter the actual IP address of your partner's gatekeeper!

3. In PortaBilling, create a new “VoIP to vendor connection” and enter **Name/IP** in the Remote IP field, where “Name” is the section name in `gateway.cfg` and “IP” is the actual address of the gatekeeper. Thus, in our example above, you will enter `GKPARTNER1/12.34.56.78`.



Edit	Load	H323	SIP	Remote ID *	Transl. Rule	Outgoing Rule	CLI Transl. Rule	Account	Capacity *	Delete	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	GKPARTNER1/12.34.56.78				None		<input type="checkbox"/>	
				GK rates	Traffic for GK-enabled termination				10		<input type="checkbox"/>



It is extremely important that you use the same value in MVTS and the PortaBilling configuration; otherwise calls may not be routed or billed properly.

Now, when a call is to be routed to this vendor, the following will happen:

4. PortaBilling will return the route information to MVTS, including the section name (GKPARTNER1) there.
5. MVTS will use the GKPARTNER section instead of the default PORTA. MVTS will then obtain additional call routing information from the corresponding gatekeeper (the one which the GKPARTNER section points to).
6. After the call has been terminated to the remote gateway, MVTS will send an accounting record to the billing. In the h323-remote-address field, however, MVTS will enter the address of the remote gatekeeper (12.34.56.78).
7. PortaBilling will then be able to match a connection to the vendor by this address, and the call will be billed properly.

6. Troubleshooting

In case of any difficulties, you can find comprehensive information in the MVTS and PortaBilling log files.

MVTS RADIUS logs

In order to monitor the RADIUS dialogue from the MVTS side, you should

Need information from Mera

Below you will find an example of the log information:

```
DEMO 12.10.05 09:22:36, (2+) ,Sent 70.68.0.213:1813 Radius
AccountingRequest {
  session id = 160
  AcctStatusType - Start
```

This is a new incoming VoIP call

```
UserName: 16044321
NasIpAddress: 216.231.44.193
NasPortType: 0
ServiceType: 1
AcctDelayTime: 0
CalledStationId: 5555555
AcctSessionId: 4e08e1ab-37-25c2b68eT2
Cisco VSA( 2): 70.68.0.213
Cisco VSA( 26): h323-call-origin=answer
Cisco VSA( 27): h323-call-type=Telephony
Cisco VSA( 1): h323-incoming-conf-id=55C6071C 55673645
80E78C54 40639934
Cisco VSA( 1): h323-incoming-call-id=1A762A51 FDE1904F
AFD1D150 F3D4BB4D
Cisco VSA( 25): h323-setup-time=09:22:35.000 PDT Wed Oct 12
2005
Cisco VSA( 33): h323-gw-id=mvts-ua
Cisco VSA( 1): h323-gw-address=70.68.0.213
Cisco VSA( 24): h323-conf-id=55C6071C 55673645 80E78C54
40639934
Cisco VSA( 23): h323-remote-address=193.28.87.3
Cisco VSA( 1): h323-remote-id=193.28.87.3
Cisco VSA( 1): xpgk-h323-id=Joe Kotnjek
Cisco VSA( 1): xpgk-dst-number-in=5555555
Cisco VSA( 1): xpgk-dst-number-out=5555555
Cisco VSA( 1): xpgk-route-retries=2
Cisco VSA( 1): h323-call-id=1A762A51 FDE1904F AFD1D150
F3D4BB4D
```

```
DEMO 12.10.05 09:22:36, (2+) ,Recv 70.68.0.213:1813 Radius
```

AccountingResponse

```
session id = 160
```

```
DEMO 12.10.05 10:49:18, (2+) ,Sent 70.68.0.213:1812 Radius
```

```
AccessRequest {
session id = 162
UserName: 16044321
```

MVTS requests user authentication

```
UserPassword: a82faa5b7bad5bc5937583f998de592c
NasIpAddress: 216.231.44.193
NasPortType: 0
ServiceType: 1
Cisco VSA( 1): xpgk-request-type=user
Cisco VSA( 1): xpgk-source-addr=70.68.0.213:1523
```

DEMO 12.10.05 10:49:18, (2+) ,Recv 70.68.0.213:1812 Radius

```
AccessAccept {
```

PortaBilling acknowledges authentication request

```
session id = 162
Cisco VSA( 1): xpgk-ep-number=90016044321
Cisco VSA(101): h323-credit-amount=0.00
Cisco VSA(109): h323-billing-model=0
Cisco VSA( 1): h323-ivr-in=Tariff:Retail Tariff
Cisco VSA( 1): h323-ivr-
in=PortaBilling_CustomerBalance:3.00000
Cisco VSA( 1): h323-ivr-in=available-funds:100.00
Cisco VSA(103): h323-return-code=0
Cisco VSA(110): h323-currency=USD
Cisco VSA(107): h323-preferred-lang=en
```

MVTS Complete Call Log Report

DEMO 25.10.05 10:26:16, (2+) ,Sent 70.68.0.213:1812 Radius
Access Request

```
session id = 9
UserName: 16044321
UserPassword: 4fab66c4564a3621813b25728b50c9d8
NasIpAddress: 216.231.44.193
NasPortType: 0
ServiceType: 1
CalledStationId: 5555555
Cisco VSA( 1): xpgk-request-type =route
Cisco VSA( 1): xpgk-routing-request
=1
Cisco VSA( 24): h323-conf-id =5C8AD575 70F0DE44 A2CED1C5 3506547F
Cisco VSA( 1): h323-call-id =6E4D5DF7 2F83CD4C 9E416ED7 175E1384
Cisco VSA( 33): h323-gw-id =mvts-ua
Cisco VSA( 1): h323-gw-address =70.68.0.213
Cisco VSA( 1): xpgk-h323-id =Joe Kotnjek
Cisco VSA( 1): xpgk-dst-number-In =5555555
Cisco VSA( 1): xpgk-dst-number-out =5555555
Cisco VSA( 1): xpgk-route-retries=1
```

DEMO 25.10.05 10:26:16, (2+) ,Recv 70.68.0.213:1812 Radius
Access Accept

```
session id = 9
Cisco VSA(109): h323-billing-model=0
Cisco VSA( 1): h323-ivr-in=Tariff:Retail Customer Tariff
```

```
Cisco VSA( 1): h323-ivr- in=PortaBilling_CompleteNumber:5555555
Cisco VSA(103): h323-return-code=13
Cisco VSA(252): PORTA/1//5555555//5555555/193.28.87.3:1720
Cisco VSA(110): h323-currency=USD
Cisco VSA(107): h323-preferred-lang=en
```

```
DEMO 25.10.05 10:26:16, (2+) ,Sent 70.68.0.213:1812 Radius
Access Request {
```

```
session id = 8
UserName: 16044321
```

```
UserPassword: 6fd75a7dfca79d14783bb5fb30ad3afb
NasIpAddress: 216.231.44.193
NasPortType: 0
ServiceType: 1
CalledStationId: 5555555
```

```
Cisco VSA( 1): xpgk-request-type=number
Cisco VSA( 24): h323-conf-id=5C8AD575 70F0DE44 A2CED1C5 3506547F
Cisco VSA( 1): h323-call-id=6E4D5DF7 2F83CD4C 9E416ED7 175E1384
Cisco VSA( 33): h323-gw-id=mvts-ua
Cisco VSA( 1): h323-gw-address=70.68.0.213
Cisco VSA( 23): h323-remote-address=193.28.87.3
Cisco VSA( 1): h323-remote-id=193.28.87.3
Cisco VSA( 1): xpgk-h323-id=Joe Kotnjek
Cisco VSA( 1): xpgk-dst-number-in=5555555
Cisco VSA( 1): xpgk-dst-number-out=5555555
Cisco VSA( 1): xpgk-route-retries=2
```

```
DEMO 25.10.05 10:26:16, (2+) ,Recv 70.68.0.213:1812 Radius
Access Accept
```

```
session id = 8
Cisco VSA(109): h323-billing-model=0
Cisco VSA( 1): h323-ivr-in=Tariff:Retail Customer Tariff
Cisco VSA(103): h323-return-code=13
Cisco VSA(110): h323-currency=USD
Cisco VSA(107): h323-preferred-lang=en
```

```
DEMO 25.10.05 10:26:16, (2+) ,Sent 70.68.0.213:1813 Radius
Accounting Request
```

```
session id = 8
AcctStatusType - Start
UserName: 16044321
```

```
NasIpAddress: 216.231.44.193
NasPortType: 0
ServiceType: 1
AcctDelayTime: 0
CalledStationId: 5555555
AcctSessionId: 973b5010-3-4a4fef0T2
Cisco VSA( 2): 70.68.0.213
Cisco VSA( 26): h323-call-origin=answer
Cisco VSA( 27): h323-call-type=Telephony
Cisco VSA( 1): h323-incoming-conf-id=5C8AD575 70F0DE44 A2CED1C5
3506547F
Cisco VSA( 1): h323-incoming-call-id=6E4D5DF7 2F83CD4C 9E416ED7
175E1384
Cisco VSA( 25): h323-setup-time=10:26:16.000 PDT Tue Oct 25 2005
Cisco VSA( 33): h323-gw-id=mvts-ua
Cisco VSA( 1): h323-gw-address=70.68.0.213
```

```
Cisco VSA( 24): h323-conf-id=5C8AD575 70F0DE44 A2CED1C5 3506547F
Cisco VSA( 23): h323-remote-address=193.28.87.3
Cisco VSA( 1): h323-remote-id=193.28.87.3
Cisco VSA( 1): xpgk-h323-id=Joe Kotnjek
Cisco VSA( 1): xpgk-dst-number-in=5555555
Cisco VSA( 1): xpgk-dst-number-out=5555555
Cisco VSA( 1): xpgk-route-retries=2
Cisco VSA( 1): h323-call-id=6E4D5DF7 2F83CD4C 9E416ED7 175E1384
```

```

DEMO 25.10.05 10:26:16, (2+) ,Recv 70.68.0.213:1813 Radius
Accounting Response
session id = 8

DEMO 25.10.05 10:26:16, (2+) ,Sent 70.68.0.213:1813 Radius
AccountingRequest {
session id = 8
AcctStatusType - Start
UserName: 16044321
NasIpAddress: 216.231.44.193
NasPortType: 0
ServiceType: 1
AcctDelayTime: 0
CalledStationId: 5555555
AcctSessionId: 973b5010-3-4a4fef0V2
Cisco VSA( 26): h323-call-origin=originate
Cisco VSA( 27): h323-call-type=VoIP
Cisco VSA( 1): h323-incoming-conf-id=5C8AD575 70F0DE44 A2CED1C5
3506547F
Cisco VSA( 1): h323-incoming-call-id=6E4D5DF7 2F83CD4C 9E416ED7
175E1384
Cisco VSA( 25): h323-setup-time=10:26:16.000 PDT Tue Oct 25 2005
Cisco VSA( 33): h323-gw-id=mvts-ua
Cisco VSA( 1): h323-gw-address=70.68.0.213
Cisco VSA( 24): h323-conf-id=5C8AD575 70F0DE44 A2CED1C5 3506547F
Cisco VSA( 23): h323-remote-address=193.28.87.3
Cisco VSA( 1): h323-remote-id=193.28.87.3
Cisco VSA( 1): xpgk-h323-id=Joe Kotnjek
Cisco VSA( 1): xpgk-dst-number-in=5555555
Cisco VSA( 1): xpgk-dst-number-out=5555555
Cisco VSA( 1): xpgk-route-retries=2
Cisco VSA( 1): h323-call-id=6E4D5DF7 2F83CD4C 9E416ED7 175E1384

DEMO 25.10.05 10:26:16, (2+) ,Recv 70.68.0.213:1813 Radius
AccountingResponse
session id = 8

DEMO 25.10.05 10:26:16, (2+) ,Sent 193.28.87.3:1720 Q.931

protocolDiscriminator = 8
callReference = 2119
from = originator
messageType = Setup
IE: Bearer-Capability = {
80 90 a3 ...
IE: Called-Party-Number = {
80 35 35 35 35 35 35 35 35 .5555555
IE: User-User = {
00 98 06 00 08 91 4a 00 02 22 c0 b5 00 53 4c 16 .....J.."...SL.
4d 69 63 72 6f 73 6f 66 74 ae 20 4e 65 74 4d 65 Microsoft. NetMe
65 74 69 6e 67 ae 00 03 33 2e 30 00 00 01 03 00 eting...3.0.....
88 88 88 80 c1 1c 57 03 06 b8 00 5c 8a d5 75 70 .....W....\..up
f0 de 44 a2 ce d1 c5 35 06 54 7f 00 45 0c 07 00 ..D....5.T..E...
d8 e7 2c c1 06 b8 11 00 6e 4d 5d f7 2f 83 cd 4c .....nM]../..L
9e 41 6e d7 17 5e 13 84 01 00 01 00 .An..^.....

UserUserField:

h323_uu_pdu = {
h323_message_body = setup {
protocolIdentifier = 0.0.8.2250.0.2
sourceInfo = {
vendor = {
vendor = {
t35CountryCode = 181

```

```

t35Extension = 0
manufacturerCode = 21324
productId = 23 octets
4d 69 63 72 6f 73 6f 66 74 ae 20 4e 65 74 4d 65  Microsoft. NetMe
65 74 69 6e 67 ae 00 eting..
versionId = 8 octets {
74 64 5a 00 02 00 00 00 tdZ.....
terminal =
mc = FALSE
undefinedNode = FALSE
destinationAddress = 1 entries {
[0]=dialedDigits "5555555"
destCallSignalAddress = ipAddress
ip = 4 octets {
c1 1c 57 03 ..W.
port = 1720
activeMC = FALSE
conferenceID = 16 octets {
5c 8a d5 75 70 f0 de 44 a2 ce d1 c5 35 06 54 7f \..up..D....5.T.
conferenceGoal = create <<null>>
callType = pointToPoint <<null>>
sourceCallSignalAddress = ipAddress {
ip = 4 octets {
d8 e7 2c c1 ....
port = 1720
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
mediaWaitForConnect = FALSE
canOverlapSend = FALSE

```

```

DEMO 25.10.05 10:26:16, (2+) ,Recv 193.28.87.3:1720
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..

```

Q.931

```

protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = SetupAck
IE: User-User = {
28 40 19 00 06 00 08 91 4a 00 04 00 6e 4d 5d f7 (@.....J...nM].
2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 10 80 01 00 /..L.An..^.....

```

UserUserField:

```

h323_uu_pdu = {
h323_message_body = setupAcknowledge {
protocolIdentifier = 0.0.8.2250.0.4
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
h245Tunneling = FALSE

```

```

DEMO 25.10.05 10:26:16, (2+) ,Recv 193.28.87.3:1720
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..

```

Q.931

```

protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = CallProceeding
IE: User-User = {
21 c0 06 00 08 91 4a 00 04 28 00 b5 00 00 12 40 !.....J..(.....@
01 3c 05 04 01 00 20 40 00 c1 1c 57 03 32 33 11 .<.... @...W.23.
0c 11 00 6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 .....nM]../..L.An..
5e 13 84 01 00 01 00 10 80 01 00 ^.....

```

UserUserField:

```

h323_uu_pdu = {
h323_message_body = callProceeding {
protocolIdentifier = 0.0.8.2250.0.4
destinationInfo = {
vendor = {
vendor = {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18

gateway = {
protocol = 1 entries {
[0]=voice {
supportedPrefixes = 1 entries {
[0]={
prefix = dialedDigits "1#"
mc = FALSE
undefinedNode = FALSE
h245Address = ipAddress {
ip = 4 octets {
c1 1c 57 03 ..W.
port = 12851
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
multipleCalls = FALSE
maintainConnection = FALSE
h245Tunneling = FALSE

```

```

DEMO 25.10.05 10:26:16, (2+) ,Recv 193.28.87.3:1720
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..

```

Q.931

```

protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = Connect
IE: Bearer-Capability = {
80 90 a3 ...
IE: User-User = {
22 c0 06 00 08 91 4a 00 04 00 c1 1c 57 03 32 33 ".....J.....W.23
28 00 b5 00 00 12 40 01 3c 05 04 01 00 20 40 00 (.....@.<.... @.
5c 8a d5 75 70 f0 de 44 a2 ce d1 c5 35 06 54 7f \..up..D....5.T.
1d 0c 00 11 00 6e 4d 5d f7 2f 83 cd 4c 9e 41 6e .....nM]../..L.An..
d7 17 5e 13 84 01 00 01 00 10 80 01 00 ..^.....

```

UserUserField:

```
h323_uu_pdu = {
h323_message_body = connect {
protocolIdentifier = 0.0.8.2250.0.4
h245Address = ipAddress {
ip = 4 octets {
c1 1c 57 03          ..W.
port = 12851
destinationInfo = {
vendor = {
vendor = {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
gateway = {
protocol = 1 entries {
[0]=voice {
supportedPrefixes = 1 entries {
[0]={
prefix = dialedDigits "1#"
mc = FALSE
undefinedNode = FALSE
conferenceID = 16 octets {
5c 8a d5 75 70 f0 de 44 a2 ce d1 c5 35 06 54 7f  \..up..D....5.T.
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84  nM]../..L.An..^..
multipleCalls = FALSE
maintainConnection = FALSE
h245Tunneling = FALSE
```

```
DEMO 25.10.05 10:26:16, (2+) ,Sent 70.68.0.213:1278
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84  nM]../..L.An..^..
Q.931
```

```
protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = SetupAck
IE: User-User = {
28 40 19 00 06 00 08 91 4a 00 04 00 6e 4d 5d f7 (@.....J...nM].
2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 10 80 01 00  /..L.An..^.....
```

UserUserField:

```
h323_uu_pdu = {
h323_message_body = setupAcknowledge {
protocolIdentifier = 0.0.8.2250.0.4
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84  nM]../..L.An..^..
h245Tunneling = FALSE
```

```
DEMO 25.10.05 10:26:16, (2+) ,Sent 70.68.0.213:1278
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84  nM]../..L.An..^..
Q.931
```

```
protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = CallProceeding
IE: User-User = {
21 c0 06 00 08 91 4a 00 04 28 00 b5 00 00 12 40  !.....J..(.....@
```

```
01 3c 05 04 01 00 20 40 00 d8 e7 2c c1 fe dd 0d .<.... @.....
0c 11 00 6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 ...nM]../..L.An..
5e 13 84 01 00 01 00 02 80 01 00 ^.....
```

UserUserField:

```
h323_uu_pdu = {
h323_message_body = callProceeding {
protocolIdentifier = 0.0.8.2250.0.4
destinationInfo = {
vendor = {
vendor = {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
gateway = {
protocol = 1 entries {
[0]=voice {
supportedPrefixes = 1 entries {
0}={
prefix = dialedDigits "1#"
mc = FALSE
undefinedNode = FALSE
h245Address = ipAddress {
ip = 4 octets {
d8 e7 2c c1 ....
port = 65245
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
multipleCalls = FALSE
maintainConnection = FALSE
h245Tunneling = FALSE
```

```
DEMO 25.10.05 10:26:16, (2+) ,Sent 70.68.0.213:1278
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
```

Q.931

```
protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = Connect
IE: Bearer-Capability = {
80 90 a3 ...
IE: User-User = {
22 c0 06 00 08 91 4a 00 04 00 d8 e7 2c c1 fe dd ".....J.....
28 00 b5 00 00 12 40 01 3c 05 04 01 00 20 40 00 (.....@.<.... @.
5c 8a d5 75 70 f0 de 44 a2 ce d1 c5 35 06 54 7f \..up..D....5.T.
0d 0c 11 00 6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 ...nM]../..L.An.
17 5e 13 84 01 00 01 00 02 80 01 00 ^.....
```

UserUserField:

```
h323_uu_pdu = {
h323_message_body = connect {
protocolIdentifier = 0.0.8.2250.0.4
h245Address = ipAddress {
ip = 4 octets {
d8 e7 2c c1 ....
port = 65245
destinationInfo = {
vendor = {
vendor = {
t35CountryCode = 181
t35Extension = 0
```

```

manufacturerCode = 18
gateway = {
  protocol = 1 entries {
    [0]=voice {
      supportedPrefixes = 1 entries {
        [0]={
          prefix = dialedDigits
          mc = FALSE
          undefinedNode = FALSE
          conferenceID = 16 octets {
            5c 8a d5 75 70 f0 de 44 a2 ce d1 c5 35 06 54 7f  \..up..D....5.T.
          }
          callIdentifier = {
            guid = 16 octets {
              6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84  nm]../..L.An..^..
            }
            multipleCalls = FALSE
            maintainConnection = FALSE
            h245Tunneling = FALSE
          }
        }
      }
    }
  }
}

```

DEMO 25.10.05 10:26:17, (2+) ,Recv 70.68.0.213:1279 H.245

```

CALLID{
  guid = 16 octets {
    6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84  nm]../..L.An..^..
  }
  request terminalCapabilitySet {
    sequenceNumber = 1
    protocolIdentifier = 0.0.8.245.0.3
    multiplexCapability = h2250Capability {
      maximumAudioDelayJitter = 60
      receiveMultipointCapability = {
        multicastCapability = FALSE
        multiUniCastConference = FALSE
        mediaDistributionCapability = 1 entries {
          [0]={
            centralizedControl = FALSE
            distributedControl = FALSE
            centralizedAudio = FALSE
            distributedAudio = FALSE
            centralizedVideo = FALSE
            distributedVideo = FALSE
            transmitMultipointCapability = {
              multicastCapability = FALSE
              multiUniCastConference = FALSE
              mediaDistributionCapability = 1 entries {
                [0]={
                  centralizedControl = FALSE
                  distributedControl = FALSE
                  centralizedAudio = FALSE
                  distributedAudio = FALSE
                  centralizedVideo = FALSE
                  distributedVideo = FALSE
                  receiveAndTransmitMultipointCapability = {
                    multicastCapability = FALSE
                    multiUniCastConference = FALSE
                    mediaDistributionCapability = 1 entries {
                      [0]={
                        centralizedControl = FALSE
                        distributedControl = FALSE
                        centralizedAudio = FALSE
                        distributedAudio = FALSE
                        centralizedVideo = FALSE
                        distributedVideo = FALSE
                        mcCapability = {
                          centralizedConferenceMC = FALSE
                          decentralizedConferenceMC = FALSE
                          rtcpVideoControlCapability = FALSE
                          mediaPacketizationCapability = {
                            h261aVideoPacketization = FALSE
                            logicalChannelSwitchingCapability = FALSE
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

t120DynamicPortCapability = FALSE
capabilityTable = 11 entries {
[0]={
capabilityTableEntryNumber = 32768
capability = nonStandard {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 66
manufacturerCode = 32896
data = 1 octets {
01
[1]={
capabilityTableEntryNumber = 11
capability = receiveAndTransmitDataApplicationCapability {
application = t120 separateLANStack <<null>>
maxBitRate = 85000
[2]={
capabilityTableEntryNumber = 1
capability = receiveAudioCapability nonStandard [Microsoft ADPCM] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 21324
data = 72 octets {
02 00 00 00 00 00 f4 01 00 00 f4 01 01 00 04 00 .....
00 00 00 00 02 00 01 00 40 1f 00 00 00 10 00 00 .....@.....
00 01 04 00 20 00 f4 01 07 00 00 01 00 00 00 02 .....
00 ff 00 00 00 00 c0 00 40 00 f0 00 00 00 cc 01 .....@.....
30 ff 88 01 18 ff 00 00 .....0.....
[3]={
capabilityTableEntryNumber = 2
capability = receiveAudioCapability g711Alaw64k 180
[4]={
capabilityTableEntryNumber = 3
capability = receiveAudioCapability g711Ulaw64k 180
[5]={
capabilityTableEntryNumber = 4
capability = receiveAudioCapability g7231 {
maxAl_sduAudioFrames = 12
silenceSuppression = FALSE
[6]={
capabilityTableEntryNumber = 6
capability = receiveVideoCapability h263VideoCapability {
sqcifMPI = 1
maxBitRate = 850
unrestrictedVector = FALSE
arithmeticCoding = FALSE
advancedPrediction = FALSE
pbFrames = FALSE
temporalSpatialTradeOffCapability = TRUE
errorCompensation = FALSE
[7]={
capabilityTableEntryNumber = 7
capability = receiveVideoCapability h263VideoCapability {
qcifMPI = 1
maxBitRate = 850
unrestrictedVector = FALSE
arithmeticCoding = FALSE
advancedPrediction = FALSE
pbFrames = FALSE
temporalSpatialTradeOffCapability = TRUE
errorCompensation = FALSE
[8]={
capabilityTableEntryNumber = 8
capability = receiveVideoCapability h263VideoCapability {
cifMPI = 1
maxBitRate = 850
unrestrictedVector = FALSE

```

```

arithmeticCoding = FALSE
advancedPrediction = FALSE
pbFrames = FALSE
temporalSpatialTradeOffCapability = TRUE
errorCompensation = FALSE
[9]={
capabilityTableEntryNumber = 9
capability = receiveVideoCapability h261VideoCapability {
qcifMPI = 1
temporalSpatialTradeOffCapability = TRUE
maxBitRate = 850
stillImageTransmission = FALSE
videoBadMBsCap = FALSE
[10]={
capabilityTableEntryNumber = 10
capability = receiveVideoCapability h261VideoCapability {
cifMPI = 1
temporalSpatialTradeOffCapability = TRUE
maxBitRate = 850
stillImageTransmission = FALSE
videoBadMBsCap = FALSE
capabilityDescriptors = 5 entries {
[0]={
capabilityDescriptorNumber = 6
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1
[1]=1 entries {
[0]=7
[2]=1 entries {
[0]=1
[1]={
capabilityDescriptorNumber = 7
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1
[1]=1 entries {
[0]=6
[2]=1 entries {
[0]=11
[2]=
capabilityDescriptorNumber = 8
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1
[ 1]=1 entries {
[0]=9
[2]=1 entries {
[0]=11
[3]=
capabilityDescriptorNumber = 9
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1
[1]=1 entries {

```

```
[0]=8
[ 2]=1 entries {
[0]=11
[4]={
capabilityDescriptorNumber = 10
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1
[1]=1 entries {
[0]=10
[2]=1 entries {
[0]=11
```

DEMO 25.10.05 10:26:17, (2+) ,Sent 193.28.87.3:12851 H.245

```
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
```

```
request terminalCapabilitySet {
sequenceNumber = 1
protocolIdentifier = 0.0.8.245.0.3
multiplexCapability = h2250Capability {
maximumAudioDelayJitter = 60
receiveMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
```

```
transmitMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
```

```
receiveAndTransmitMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
```

```

centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE

mcCapability =
centralizedConferenceMC = FALSE
decentralizedConferenceMC = FALSE

rtcpVideoControlCapability = FALSE
mediaPacketizationCapability = {
h261aVideoPacketization = FALSE

logicalChannelSwitchingCapability = FALSE
t120DynamicPortCapability = FALSE

capabilityTable = 11 entries {
[ 0]={
capabilityTableEntryNumber = 32768
capability = nonStandard {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 66
manufacturerCode = 32896

data = 1 octets {
01

[1]={
capabilityTableEntryNumber = 11
capability = receiveAndTransmitDataApplicationCapability {
application = t120 separateLANStack <<null>>
maxBitRate = 85000

[2]={
capabilityTableEntryNumber = 1
capability = receiveAudioCapability nonStandard [Microsoft ADPCM] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 21324
data = 72 octets {
02 00 00 00 00 00 f4 01 00 00 f4 01 01 00 04 00 .....
00 00 00 00 02 00 01 00 40 1f 00 00 00 10 00 00 .....@.....
00 01 04 00 20 00 f4 01 07 00 00 01 00 00 00 02 ....
00 ff 00 00 00 00 c0 00 40 00 f0 00 00 00 cc 01 .....@.....
30 ff 88 01 18 ff 00 00 0.....

[3]={
capabilityTableEntryNumber = 2
capability = receiveAudioCapability g711Alaw64k 180

[4]={
capabilityTableEntryNumber = 3
capability = receiveAudioCapability g711Ulaw64k 180

[5]={
capabilityTableEntryNumber = 4
capability = receiveAudioCapability g7231 {
maxAl_sduAudioFrames = 12
silenceSuppression = FALSE

[6]={
capabilityTableEntryNumber = 6
capability = receiveVideoCapability h263VideoCapability {

```

```
sqcifMPI = 1
maxBitRate = 850
unrestrictedVector = FALSE
arithmeticCoding = FALSE
advancedPrediction = FALSE
pbFrames = FALSE
temporalSpatialTradeOffCapability = TRUE
errorCompensation = FALSE

[7]={
capabilityTableEntryNumber = 7
capability = receiveVideoCapability h263VideoCapability {
qcifMPI = 1
maxBitRate = 850
unrestrictedVector = FALSE
arithmeticCoding = FALSE
advancedPrediction = FALSE
pbFrames = FALSE
temporalSpatialTradeOffCapability = TRUE
errorCompensation = FALSE

[8]={
capabilityTableEntryNumber = 8
capability = receiveVideoCapability h263VideoCapability {
cifMPI = 1
maxBitRate = 850
unrestrictedVector = FALSE
arithmeticCoding = FALSE
advancedPrediction = FALSE
pbFrames = FALSE
temporalSpatialTradeOffCapability = TRUE
errorCompensation = FALSE

[9]={
capabilityTableEntryNumber = 9
capability = receiveVideoCapability h261VideoCapability {
qcifMPI = 1
temporalSpatialTradeOffCapability = TRUE
maxBitRate = 850
stillImageTransmission = FALSE
videoBadMBSCap = FALSE

[10]={
capabilityTableEntryNumber = 10
capability = receiveVideoCapability h261VideoCapability {
cifMPI = 1
temporalSpatialTradeOffCapability = TRUE
maxBitRate = 850
stillImageTransmission = FALSE
videoBadMBSCap = FALSE

capabilityDescriptors = 5 entries {
[0]={
capabilityDescriptorNumber = 6
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1

[1]=1 entries {
```

```
[0]=7

[2]=1 entries {
[0]=11

[1]={
capabilityDescriptorNumber = 7
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1

[1]=1 entries {
[0]=6

[2]=1 entries {
[0]=11
[2]={
capabilityDescriptorNumber = 8
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1

[1]=1 entries {
[0]=9

[2]=1 entries {
[0]=11

[3]={
capabilityDescriptorNumber = 9
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1

[1]=1 entries {
[0]=8

[2]=1 entries {
[0]=11

[4]={
capabilityDescriptorNumber = 10
simultaneousCapabilities = 3 entries {
[0]=4 entries {
[0]=4
[1]=3
[2]=2
[3]=1

[1]=1 entries {
[0]=10

[2]=1 entries {
[0]=11
```

```
DEMO 25.10.05 10:26:17, (2+) ,Recv 70.68.0.213:1279 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
request masterSlaveDetermination {
terminalType = 50
statusDeterminationNumber = 5998640
```

```
DEMO 25.10.05 10:26:17, (2+) ,Sent 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
request masterSlaveDetermination {
terminalType = 50
statusDeterminationNumber = 5998640
```

```
DEMO 25.10.05 10:26:17, (2+) ,Recv 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
request terminalCapabilitySet {
sequenceNumber = 1
protocolIdentifier = 0.0.8.245.0.7
multiplexCapability = h2250Capability {
maximumAudioDelayJitter = 20
receiveMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
transmitMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
receiveAndTransmitMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
mcCapability = {
centralizedConferenceMC = FALSE
decentralizedConferenceMC = FALSE
rtcpVideoControlCapability = FALSE
```

```
mediaPacketizationCapability = {
h261aVideoPacketization = FALSE
logicalChannelSwitchingCapability = FALSE
t120DynamicPortCapability = FALSE
capabilityTable = 17 entries {
[0]={
capabilityTableEntryNumber = 30
capability = receiveRTPAudioTelephonyEventCapability {
dynamicRTPPayloadType = 101
audioTelephoneEvent = "0-16"
[1]={
capabilityTableEntryNumber = 20
capability = receiveAndTransmitDataApplicationCapability {
application = nonStandard {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18

data = 12 octets {
52 74 70 44 74 6d 66 52 65 6c 61 79          RtpDtmfRelay
maxBitRate = 0
[2]={
capabilityTableEntryNumber = 26
capability = receiveUserInputCapability hookflash <<null>>

[3]={
capabilityTableEntryNumber = 25
capability = receiveUserInputCapability dtmf <<null>>

[4]={
capabilityTableEntryNumber = 22
capability = receiveUserInputCapability basicString <<null>>

[5]={
capabilityTableEntryNumber = 4
capability = receiveAudioCapability g729AnnexA 2

[6]={
capabilityTableEntryNumber = 3
capability = receiveAudioCapability g729 2

[7]={
capabilityTableEntryNumber = 11
capability = receiveAudioCapability g729wAnnexB 2

[8]={
capabilityTableEntryNumber = 12
capability = receiveAudioCapability g729AnnexAwAnnexB 2
[9]={
capabilityTableEntryNumber = 13
capability = receiveAudioCapability g7231 {
maxAl_sduAudioFrames = 1
silenceSuppression = TRUE
[10]={
capabilityTableEntryNumber = 14
capability = receiveAudioCapability nonStandard [Cisco G7231ar] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18

data = 7 octets {
47 37 32 33 31 61 72          G7231ar
[11]={
capabilityTableEntryNumber = 9
capability = receiveAudioCapability g7231 {
maxAl_sduAudioFrames = 1
```

```
silenceSuppression = FALSE
[12]={
capabilityTableEntryNumber = 1
capability = receiveAudioCapability g711Ulaw64k 20

[13]={
capabilityTableEntryNumber = 2
capability = receiveAudioCapability g711Alaw64k 20

[14]={
capabilityTableEntryNumber = 7
capability = receiveAudioCapability nonStandard [Cisco G726r32] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 7 octets {
47 37 32 36 72 33 32                                G726r32
[15]={
capabilityTableEntryNumber = 6
capability = receiveAudioCapability nonStandard [Cisco G726r24] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 7 octets {
47 37 32 36 72 32 34                                G726r24

[16]={
capabilityTableEntryNumber = 5
capability = receiveAudioCapability nonStandard [Cisco G726r16] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 7 octets {
47 37 32 36 72 31 36                                G726r16

capabilityDescriptors = 1 entries {
[0]={
capabilityDescriptorNumber = 1
simultaneousCapabilities = 3 entries {
[0]=12 entries {
[0]=3
[1]=4
[2]=9
[3]=11
[4]=12
[5]=5
[6]=6
[7]=7
[8]=2
[9]=1
[10]=14
[11]=13

[1]=4 entries {
[0]=30
[1]=25
[2]=22
[3]=20

[2]=1 entries {
[0]=26
```

```
DEMO 25.10.05 10:26:17, (2+) ,Sent 70.68.0.213:1279 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nm]../..L.An..^..
request terminalCapabilitySet {
sequenceNumber = 1
protocolIdentifier = 0.0.8.245.0.7
multiplexCapability = h2250Capability {
maximumAudioDelayJitter = 20
receiveMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
transmitMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
receiveAndTransmitMultipointCapability = {
multicastCapability = FALSE
multiUniCastConference = FALSE
mediaDistributionCapability = 1 entries {
[0]={
centralizedControl = FALSE
distributedControl = FALSE
centralizedAudio = FALSE
distributedAudio = FALSE
centralizedVideo = FALSE
distributedVideo = FALSE
mcCapability = {
centralizedConferenceMC = FALSE
decentralizedConferenceMC = FALSE
rtcpVideoControlCapability = FALSE
mediaPacketizationCapability = {
h261aVideoPacketization = FALSE
logicalChannelSwitchingCapability = FALSE
t120DynamicPortCapability = FALSE
capabilityTable = 16 entries {
[0]={
capabilityTableEntryNumber = 20
capability = receiveAndTransmitDataApplicationCapability {
application = nonStandard {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 12 octets {
52 74 70 44 74 6d 66 52 65 6c 61 79 RtpDtmfRelay
maxBitRate = 0
[1]={
capabilityTableEntryNumber = 26
capability = receiveUserInputCapability hookflash <<null>>
```

```
[2]={
capabilityTableEntryNumber = 25
capability = receiveUserInputCapability dtmf <<null>>

[3]={
capabilityTableEntryNumber = 22
capability = receiveUserInputCapability basicString <<null>>

[4]={
capabilityTableEntryNumber = 4
capability = receiveAudioCapability g729AnnexA 2

[5]={
capabilityTableEntryNumber = 3
capability = receiveAudioCapability g729 2

[6]={
capabilityTableEntryNumber = 11
capability = receiveAudioCapability g729wAnnexB 2

[7]={
capabilityTableEntryNumber = 12
capability = receiveAudioCapability g729AnnexAwAnnexB 2

[8]={
capabilityTableEntryNumber = 13
capability = receiveAudioCapability g7231 {
maxAl_sduAudioFrames = 1
silenceSuppression = TRUE

[9]={
capabilityTableEntryNumber = 14
capability = receiveAudioCapability nonStandard [Cisco G7231ar] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 7 octets {
47 37 32 33 31 61 72                                G7231ar

[10]={
capabilityTableEntryNumber = 9
capability = receiveAudioCapability g7231 {
maxAl_sduAudioFrames = 1
silenceSuppression = FALSE

[11]={
capabilityTableEntryNumber = 1
capability = receiveAudioCapability g711Ulaw64k 20

[12]={
capabilityTableEntryNumber = 2
capability = receiveAudioCapability g711Alaw64k 20

[13]={
capabilityTableEntryNumber = 7
capability = receiveAudioCapability nonStandard [Cisco G726r32] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 7 octets {
47 37 32 36 72 33 32                                G726r32

[14]={
capabilityTableEntryNumber = 6
capability = receiveAudioCapability nonStandard [Cisco G726r24] {
```

```

nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 7 octets {
47 37 32 36 72 32 34                                     G726r24

[15]={
capabilityTableEntryNumber = 5
capability = receiveAudioCapability nonStandard [Cisco G726r16] {
nonStandardIdentifier = h221NonStandard {
t35CountryCode = 181
t35Extension = 0
manufacturerCode = 18
data = 7 octets {
47 37 32 36 72 31 36                                     G726r16

capabilityDescriptors = 1 entries {
[0]={
capabilityDescriptorNumber = 1
simultaneousCapabilities = 3 entries {
[0]=12 entries {
[0]=3
[1]=4
[2]=9
[3]=11
[4]=12
[5]=5
[6]=6
[7]=7
[8]=2
[9]=1
[10]=14
[11]=13
[1]=3 entries {
[0]=25
[1]=22
[2]=20
[2]=1 entries {
[0]=26

```

DEMO 25.10.05 10:26:17, (2+) ,Recv 193.28.87.3:12851 H.245

```

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..

```

```

request masterSlaveDetermination {
terminalType = 60
statusDeterminationNumber = 1372

```

DEMO 25.10.05 10:26:17, (2+) ,Sent 70.68.0.213:1279 H.245

```

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
request masterSlaveDetermination {
terminalType = 60
statusDeterminationNumber = 1372

```

```
DEMO 25.10.05 10:26:17, (2+) ,Recv 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response terminalCapabilitySetAck {
sequenceNumber = 1

DEMO 25.10.05 10:26:17, (2+) ,Sent 70.68.0.213:1279 H.245
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response terminalCapabilitySetAck {
sequenceNumber = 1

DEMO 25.10.05 10:26:17, (2+) ,Recv 70.68.0.213:1279 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response terminalCapabilitySetAck {
sequenceNumber = 1

DEMO 25.10.05 10:26:17, (2+) ,Sent 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response terminalCapabilitySetAck {
sequenceNumber = 1

DEMO 25.10.05 10:26:17, (2+) ,Recv 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response masterSlaveDeterminationAck {
decision = slave <<null>>

DEMO 25.10.05 10:26:17, (2+) ,Sent 70.68.0.213:1279 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response masterSlaveDeterminationAck {
decision = slave <<null>>

DEMO 25.10.05 10:26:17, (2+) ,Recv 70.68.0.213:1279 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response masterSlaveDeterminationAck {
decision = master <<null>>

DEMO 25.10.05 10:26:17, (2+) ,Sent 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]../..L.An..^..
response masterSlaveDeterminationAck {
```

```
decision = master <<null>>

DEMO 25.10.05 10:26:17, (2+) ,Recv 70.68.0.213:1279 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
request openLogicalChannel {
forwardLogicalChannelNumber = 257
forwardLogicalChannelParameters = {
dataType = audioData g7231 {
maxAL_sduAudioFrames = 1
silenceSuppression = FALSE

multiplexParameters = h2250LogicalChannelParameters {

sessionID = 1
mediaGuaranteedDelivery = FALSE
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
c0 a8 00 5a ...Z

tsapIdentifier = 49605

mediaControlGuaranteedDelivery = FALSE
silenceSuppression = TRUE

DEMO 25.10.05 10:26:17, (2+) ,Sent 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
request openLogicalChannel {
forwardLogicalChannelNumber = 257
forwardLogicalChannelParameters = {
dataType = audioData g7231 {
maxAL_sduAudioFrames = 1
silenceSuppression = FALSE

multiplexParameters = h2250LogicalChannelParameters {
sessionID = 1
mediaGuaranteedDelivery = FALSE
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
d8 e7 2c c1 ....
tsapIdentifier = 16393
mediaControlGuaranteedDelivery = FALSE
silenceSuppression = TRUE

DEMO 25.10.05 10:26:18, (2+) ,Recv 193.28.87.3:12851 H.245
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
request openLogicalChannel {
forwardLogicalChannelNumber = 1
forwardLogicalChannelParameters = {
dataType = audioData g7231 {
maxAL_sduAudioFrames = 1
silenceSuppression = FALSE
multiplexParameters = h2250LogicalChannelParameters {
sessionID = 1
```

```
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
c1 1c 57 03                ..W.
tsapIdentifier = 18477
silenceSuppression = FALSE
```

DEMO 25.10.05 10:26:18, (2+) ,Sent 70.68.0.213:1279 H.245

```
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nm]../..L.An..^..
request openLogicalChannel {
forwardLogicalChannelNumber = 1
forwardLogicalChannelParameters = {
dataType = audioData g7231 {
maxAl_sduAudioFrames = 1
silenceSuppression = FALSE
ltiplexParameters = h2250LogicalChannelParameters {
sessionID = 1
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
d8 e7 2c c1
tsapIdentifier = 16395
silenceSuppression = FALSE
```

DEMO 25.10.05 10:26:18, (2+) ,Recv 193.28.87.3:12851 H.245

```
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nm]../..L.An..^..
response openLogicalChannelAck {
forwardLogicalChannelNumber = 257
forwardMultiplexAckParameters = h2250LogicalChannelAckParameters {
mediaChannel = unicastAddress ipAddress {
network = 4 octets {
c1 1c 57 03                ..W.
tsapIdentifier = 18476
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
c1 1c 57 03                ..W.
tsapIdentifier = 18477
flowControlToZero = FALSE
```

DEMO 25.10.05 10:26:18, (2+) ,Sent 70.68.0.213:1279 H.245

```
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nm]../..L.An..^..
response openLogicalChannelAck {
forwardLogicalChannelNumber = 257
forwardMultiplexAckParameters = h2250LogicalChannelAckParameters {
mediaChannel = unicastAddress ipAddress {
network = 4 octets {d8 e7 2c c1
....
tsapIdentifier = 16394
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
d8 e7 2c c1                ....
tsapIdentifier = 16395
flowControlToZero = FALSE
```

```

DEMO 25.10.05 10:26:18, (2+) ,Recv 70.68.0.213:1279 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
response openLogicalChannelAck {
forwardLogicalChannelNumber = 1
forwardMultiplexAckParameters = h2250LogicalChannelAckParameters {
sessionID = 1
mediaChannel = unicastAddress ipAddress {
network = 4 octets {
c0 a8 00 5a ...Z
tsapIdentifier = 49604
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
c0 a8 00 5a ...Z
tsapIdentifier = 49605
flowControlToZero = FALSE

```

```

DEMO 25.10.05 10:26:18, (2+) ,Sent 193.28.87.3:12851 H.245

CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
response openLogicalChannelAck {
forwardLogicalChannelNumber = 1
forwardMultiplexAckParameters = h2250LogicalChannelAckParameters {
sessionID = 1
mediaChannel = unicastAddress ipAddress {
network = 4 octets {
d8 e7 2c c1 ....
tsapIdentifier = 16392
mediaControlChannel = unicastAddress ipAddress {
network = 4 octets {
d8 e7 2c c1 ....
tsapIdentifier = 16393
flowControlToZero = FALSE

```

```

DEMO 25.10.05 10:26:20, (2+) ,Recv 193.28.87.3:1720
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..

Q.931
protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = ReleaseComplete
IE: Cause - Normal call clearing = {
80 90 ..
IE: User-User = {
25 80 06 00 08 91 4a 00 04 11 00 11 00 6e 4d 5d %.....J.....nM]
f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 10 80 01 ./...L.An..^.....
00 .

```

```

UserUserField: ....

h323_uu_pdu = {
h323_message_body = releaseComplete {

```

```
protocolIdentifier = 0.0.8.2250.0.4
callIdentifier = {
  guid = 16 octets {
    6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
```

```
h245Tunneling = FALSE
```

```
DEMO 25.10.05 10:26:20, (2+) ,Sent 193.28.87.3:1720
CALLID{
  guid = 16 octets {
    6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
```

```
Q.931
protocolDiscriminator = 8
callReference = 2119
from = originator
messageType = ReleaseComplete
IE: Cause - Normal call clearing = {
  80 90 ..
IE: User-User = {
  05 80 06 00 08 91 4a 00 02 01 11 00 6e 4d 5d f7 .....J.....nM].
  2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 /...L.An..^..
```

```
UserUserField:
```

```
h323_uu_pdu = {
  h323_message_body = releaseComplete {
    protocolIdentifier = 0.0.8.2250.0.2
    callIdentifier = {
      guid = 16 octets {
        6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
```

```
DEMO 25.10.05 10:26:20, (2+) ,Sent 70.68.0.213:1813 Radius
```

```
AccountingRequest {
  session id = 8
  AcctStatusType - Stop
  UserName: 16044321
  NasIpAddress: 216.231.44.193
  NasPortType: 0
  ServiceType: 1
  AcctInputPackets: 592
  AcctOutputPackets: 4369
  AcctInputOctets: 4684
  AcctOutputOctets: 214
  CalledStationId: 5555555
  AcctDelayTime: 0
  AcctSessionTime: 4
  AcctSessionId: 973b5010-3-4a4fef0V2
  Cisco VSA( 26): h323-call-origin=originate
  Cisco VSA( 27): h323-call-type=VoIP
  Cisco VSA( 1): h323-incoming-conf-id=5C8AD575 70F0DE44 A2CED1C5
  3506547F
  Cisco VSA( 1): h323-incoming-call-id=6E4D5DF7 2F83CD4C 9E416ED7
  175E1384
  Cisco VSA( 25): h323-setup-time=10:26:16.000 PDT Tue Oct 25 2005
  Cisco VSA( 33): h323-gw-id=mvts-ua
  Cisco VSA( 1): h323-gw-address=70.68.0.213
  Cisco VSA( 28): h323-connect-time=10:26:16.000 PDT Tue Oct 25 2005
  Cisco VSA( 29): h323-disconnect-time=10:26:20.000 PDT Tue Oct 25 2005
  Cisco VSA( 30): h323-disconnect-cause=10
  Cisco VSA( 31): h323-voice-quality=0
  Cisco VSA( 1): xpgk-local-disconnect-cause=2
```

```
Cisco VSA( 1): xpgk-src-codec=g7231
Cisco VSA( 1): xpgk-dst-codec=g7231
Cisco VSA( 1): xpgk-initial-incoming-local-address=216.231.44.193
Cisco VSA( 1): xpgk-selected-incoming-local-address=216.231.44.193
Cisco VSA( 1): xpgk-outgoing-local-address=216.231.44.193
Cisco VSA( 1): xpgk-scd-time=0
Cisco VSA( 24): h323-conf-id=5C8AD575 70F0DE44 A2CED1C5 3506547F
Cisco VSA( 23): h323-remote-address=193.28.87.3
Cisco VSA( 1): h323-remote-id=193.28.87.3
Cisco VSA( 1): xpgk-h323-id=Joe Kotnjek
Cisco VSA( 1): xpgk-dst-number-in=5555555
Cisco VSA( 1): xpgk-dst-number-out=5555555
Cisco VSA( 1): xpgk-route-retries=2
Cisco VSA( 1): h323-call-id=6E4D5DF7 2F83CD4C 9E416ED7 175E1384
```

```
DEMO 25.10.05 10:26:20, (2+) ,Sent 70.68.0.213:1278
CALLID{
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
```

Q.931

```
protocolDiscriminator = 8
callReference = 2119
from = destination
messageType = ReleaseComplete
IE: Cause - Normal call clearing = {
80 90 ..
IE: User-User = {
05 80 06 00 08 91 4a 00 02 01 11 00 6e 4d 5d f7 .....J.....nM].
2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 /...L.An..^..
```

UserUserField:

```
h323_uu_pdu = {
h323_message_body = releaseComplete {
protocolIdentifier = 0.0.8.2250.0.2
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
```

```
DEMO 25.10.05 10:26:20, (2+) ,Recv 70.68.0.213:1271 H.225
disengageRequest {
requestSeqNum = 9
endpointIdentifier = 8 characters {
0031 0036 0030 0034 0034 0033 0032 0031 16044321
conferenceID = 16 octets {
5c 8a d5 75 70 f0 de 44 a2 ce d1 c5 35 06 54 7f \..up..D....5.T.
callReferenceValue = 2119
disengageReason = normalDrop <<null>>
callIdentifier = {
guid = 16 octets {
6e 4d 5d f7 2f 83 cd 4c 9e 41 6e d7 17 5e 13 84 nM]...L.An..^..
answeredCall = FALSE
```

```
DEMO 25.10.05 10:26:20, (2+) ,Sent 70.68.0.213:1271 H.225
disengageConfirm {
requestSeqNum = 9
```

```
DEMO 25.10.05 10:26:20, (2+) ,Recv 70.68.0.213:1813 Radius
AccountingResponse {
session id = 8
```

```
DEMO 25.10.05 10:26:20, (2+) ,Sent 70.68.0.213:1813 Radius
```

```
AccountingRequest {
  session id = 8
  AcctStatusType - Stop
  UserName: 16044321
  NasIpAddress: 216.231.44.193
  NasPortType: 0
  ServiceType: 1
  AcctInputPackets: 592
  AcctOutputPackets: 4369
  AcctInputOctets: 4684
  AcctOutputOctets: 214
  CalledStationId: 5555555
  AcctDelayTime: 0
  AcctSessionTime: 4
  AcctSessionId: 973b5010-3-4a4fef0T2
  Cisco VSA( 2): 70.68.0.213
  Cisco VSA( 26): h323-call-origin=answer
  Cisco VSA( 27): h323-call-type=Telephony
  Cisco VSA( 1): h323-incoming-conf-id=5C8AD575 70F0DE44 A2CED1C5
  3506547F
  Cisco VSA( 1): h323-incoming-call-id=6E4D5DF7 2F83CD4C 9E416ED7
  175E1384
  Cisco VSA( 25): h323-setup-time=10:26:16.000 PDT Tue Oct 25 2005
  Cisco VSA( 33): h323-gw-id=mvts-ua
  Cisco VSA( 1): h323-gw-address=70.68.0.213
  Cisco VSA( 28): h323-connect-time=10:26:16.000 PDT Tue Oct 25 2005
  Cisco VSA( 29): h323-disconnect-time=10:26:20.000 PDT Tue Oct 25 2005
  Cisco VSA( 30): h323-disconnect-cause=10
  Cisco VSA( 31): h323-voice-quality=0
  Cisco VSA( 1): xpgk-local-disconnect-cause=2
  Cisco VSA( 1): xpgk-src-codec=g7231
  Cisco VSA( 1): xpgk-dst-codec=g7231
  Cisco VSA( 1): xpgk-initial-incoming-local-address=216.231.44.193
  Cisco VSA( 1): xpgk-selected-incoming-local-address=216.231.44.193
  Cisco VSA( 1): xpgk-outgoing-local-address=216.231.44.193
  Cisco VSA( 1): xpgk-scd-time=0
  Cisco VSA( 24): h323-conf-id=5C8AD575 70F0DE44 A2CED1C5 3506547F
  Cisco VSA( 23): h323-remote-address=193.28.87.3
  Cisco VSA( 1): h323-remote-id=193.28.87.3
  Cisco VSA( 1): xpgk-h323-id=Joe Kotnjek
  Cisco VSA( 1): xpgk-dst-number-in=5555555
  Cisco VSA( 1): xpgk-dst-number-out=5555555
  Cisco VSA( 1): xpgk-route-retries=2
  Cisco VSA( 1): h323-call-id=6E4D5DF7 2F83CD4C 9E416ED7 175E1384

DEMO 25.10.05 10:26:20, (2+) ,Recv 70.68.0.213:1813 Radius
AccountingResponse {
  session id = 8

DEMO 25.10.05 10:53:15, (2+) ,Sent 70.68.0.213:1271 H.225
unregistrationRequest {
  requestSeqNum = 1
  callSignalAddress = 1 entries {
  [0]=ipAddress {
  ip = 4 octets {
  46 44 00 d5 FD..
  port = 1720
  endpointIdentifier = 8 characters {
  0031 0036 0030 0034 0034 0033 0032 0031 16044321
  gatekeeperIdentifier = 4 characters {
  006d 0065 0072 0061 mera

  reason = reregistrationRequired <<null>>
```

PortaBilling log information for a call

PortaBilling log files are located in the `/var/log/porta-billing.log` file. You may browse this file with utilities such as `less`, or use the log browser on the PortaBilling web interface.

```
Oct 13 09:09:57: Processing request (BE ver1.218.2.16,pid56385):
NAS-IP-Address      = '216.231.44.193'
```

MVTS gatekeeper IP address

```
NAS-Port-Type      = 'Async'
User-Name          = '16044321'
```

Account ID

```
Called-Station-Id  = '5555555'
Service-Type       = 'Login-User'
h323-gw-id         = 'mvts-ua'
h323-conf-id       = '0CACDD5 EED3CF4C 8B1E7138 60F09C0B'
Password           = 'test123'
```

Password

```
xpgk-request-type  = 'route'
xpgk-routing-request = '1'
h323-call-id       = '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'
h323-gw-address    = '70.68.0.213'
xpgk-h323-id       = 'Joe Kotnjek'
xpgk-dst-number-in = '5555555'
xpgk-dst-number-out = '5555555'
xpgk-route-retries = '1'
```

```
Oct 13 09:09:57: This call belongs to the environment #220 'mvts_test'
Oct 13 09:09:57: PrepareNexecute 'AccountAuth'
Oct 13 09:09:57: Found Account:
16044321[161838,credit,balance=0.00000,limit=100.00000] of customer
VOIP Networks[639,balance=3.00000,limit=none]
Oct 13 09:09:57: Account 16044321 is not logged in yet
Oct 13 09:09:57: PrepareNexecute 'AccountAuth'
Oct 13 09:09:57: CLD '5555555' is not recognized as our account
Oct 13 09:09:57: Compute maximum call duration for account '16044321'
with 100 funds available ...
Oct 13 09:09:57: Checked 2005-10-13 10:09:57 America/Vancouver against
'hr{0-22} wd{su sa}, hr{23}min{0-54} wd{su sa}, hr{18-7
} wd{mo-fr}': 0
Oct 13 09:09:57: Start of call is peak level 0
Oct 13 09:09:57: PrepareNexecute 'GetPricePerDestination*'
Oct 13 09:09:57: Maximum call duration: unlimited announced as
unlimited (+0-1s...+0%) by rate 451039
Oct 13 09:09:57: Remote termination 'MVTS': Calculating routing for
5555555
Oct 13 09:09:57: Default RTP Proxy remote (routed) policy 'all',
Calling party NOT behind NAT.
Oct 13 09:09:57: Looking up routes to '5555555' using '<Default System
Routing>' routing plan
Oct 13 09:09:57: PrepareNexecute 'GetRoutingPerDestination*'
Oct 13 09:09:57: Using peak rate, since no off-peak is defined
Oct 13 09:09:57: Using peak routing plan & preferences
Oct 13 09:09:57: Using peak rate for route cost
```

```
Oct 13 09:09:57: Result routes for 5555555:  
'routes to begemot' - VoIP to the remote GW 193.28.87.3, priority=0010  
cost=0  
Oct 13 09:09:57: Logging in account '16044321'(161838) to '0CACDDD5  
EED3CF4C 8B1E7138 60F09C0B'  
Oct 13 09:09:57: Authentication acknowledge response
```

Authentication accepted

```
h323-billing-model = 0  
h323-ivr-in        = Tariff:Retail Tariff  
h323-ivr-in        = PortaBilling_CompleteNumber:5555555  
h323-return-code = 13  
xpgk-routing-reply = PORTA/1//5555555//5555555/193.28.87.3:1720
```

IP address and port of the remote gateway to which the call will be routed

```
h323-currency      = USD  
h323-preferred-lang = en  
Oct 13 09:09:57: ...Done  
  
Oct 13 09:09:57: Processing request (BE ver1.218.2.16,pid56385):
```

Authorization Request

```
NAS-IP-Address      = '216.231.44.193'  
NAS-Port-Type       = 'Async'  
User-Name           = '16044321'  
Called-Station-Id   = '5555555'  
Service-Type        = 'Login-User'  
h323-gw-id          = 'mvts-ua'  
h323-conf-id        = '0CACDDD5 EED3CF4C 8B1E7138 60F09C0B'  
Password            = 'test123'  
xpgk-request-type   = 'number'  
h323-call-id        = '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'  
h323-gw-address     = '70.68.0.213'  
h323-remote-address = '193.28.87.3'  
h323-remote-id      = '193.28.87.3'  
xpgk-h323-id        = 'Joe Kotnjek'  
xpgk-dst-number-in  = '5555555'  
xpgk-dst-number-out = '5555555'  
xpgk-route-retries  = '2'  
Oct 13 09:09:57: This call belongs to the environment #220 'mvts_test'  
Oct 13 09:09:57: h323-conf-id=0CACDDD5 EED3CF4C 8B1E7138 60F09C0B/220,  
call-id=  
Oct 13 09:09:57: Found a call in cache with such id  
Oct 13 09:09:57: Copied account:  
16044321[161838,credit,balance=0.00000,limit=100.00000] of customer  
VOIP Networks[639,balance=3.00000,limit=none] from '216.231.44.193'  
into the current request  
Oct 13 09:09:57: Checking if this call comes through a VoIP from vendor  
connection  
Oct 13 09:09:57: No VoIP from vendor connections were found  
Oct 13 09:09:57: Account 16044321 is already logged in, same session  
Oct 13 09:09:57: Compute maximum call duration for account '16044321'  
with 100 funds available ...  
Oct 13 09:09:57: Checked 2005-10-13 10:09:57 America/Vancouver against  
'hr{0-22} wd{su sa}, hr{23}min{0-54} wd{su sa}, hr{18-7} wd{mo-fr}': 0  
Oct 13 09:09:57: Start of call is peak level 0  
Oct 13 09:09:57: PrepareNexecute 'GetPricePerDestination*'  
Oct 13 09:09:57: Maximum call duration: unlimited announced as  
unlimited (+0-1s...+0%) by rate 451039
```

```
Oct 13 09:09:57: Enhance lifetime with 1800s to Thu Oct 13 09:39:57
2005
Oct 13 09:09:57: Authentication acknowledge response
h323-billing-model      = 0
h323-ivr-in            = Tariff:Retail Tariff
h323-return-code       = 13
h323-currency          = USD
h323-preferred-lang    = en

Oct 13 09:09:57: ...Done
```

Account Start/ VOIP from the MVTS server

```
Oct 13 09:09:58: Processing request (BE ver1.218.2.16,pid56385):
NAS-IP-Address         = '216.231.44.193'
NAS-Port-Name          = '70.68.0.213'
NAS-Port-Type          = 'Async'
User-Name              = '16044321'
Called-Station-Id     = '5555555'
Acct-Status-Type      = 'Start'
Service-Type          = 'Login-User'
h323-gw-id            = 'mvts-ua'
h323-call-origin      = 'answer'
h323-call-type        = 'Telephony'
h323-setup-time       = '10:09:57.000 PDT Thu Oct 13 2005'
h323-conf-id          = '0CACDDD5 EED3CF4C 8B1E7138 60F09C0B'
Acct-Session-Id      = '8bc128c6-0-85339826T2'
Acct-Delay-Time       = '0'
h323-incoming-conf-id = '0CACDDD5 EED3CF4C 8B1E7138 60F09C0B'
h323-incoming-call-id = '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'
h323-gw-address       = '70.68.0.213'
h323-remote-address   = '193.28.87.3'
h323-remote-id        = '193.28.87.3'
xpgk-h323-id          = 'Joe Kotnjek'
xpgk-dst-number-in    = '5555555'
xpgk-dst-number-out   = '5555555'
xpgk-route-retries    = '2'
h323-call-id          = '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'
Exec-Program-Log      = 'porta-billing.pl'
Oct 13 09:09:58: This call belongs to the environment #220 'mvts_test'
Oct 13 09:09:58: Copied account:
16044321[161838,credit,balance=0.00000,limit=100.00000] of customer
VOIP Networks[639,balance=3.00000,limit=none] from '216.231.44.193'
into the current request
Oct 13 09:09:58: PrepareNexecute 'GetActiveLegIdByAcct'
Oct 13 09:09:58: Looking up vendor/connection
Oct 13 09:09:58: Trying to match connection for call
Oct 13 09:09:58: Looking for a connection Telephony/answer
Oct 13 09:09:58: No original CLD, using CLD from the request
Oct 13 09:09:58: Telephony, matching by the node IP '216.231.44.193',
port '70.68.0.213' and CLD '5555555'
Oct 13 09:09:58: Unknown node IP or no telephony connections for this
node
Oct 13 09:09:58: Connection to vendor not found
Oct 13 09:09:58: PrepareNexecute 'InsertActiveLeg'
Oct 13 09:09:58: Accounting response
Oct 13 09:09:58: ...Done

Oct 13 09:09:58: Processing request (BE ver1.218.2.16,pid56385):
NAS-IP-Address         = '216.231.44.193'
NAS-Port-Type          = 'Async'
User-Name              = '16044321'
Called-Station-Id     = '5555555'
Acct-Status-Type      = 'Start'
Service-Type          = 'Login-User'
h323-gw-id            = 'mvts-ua'
```

```
h323-call-origin      = 'originate'
h323-call-type        = 'VoIP'
h323-setup-time       = '10:09:57.000 PDT Thu Oct 13 2005'
h323-conf-id          = '0CACDDD5 EED3CF4C 8B1E7138 60F09C0B'
Acct-Session-Id      = '8bc128c6-0-85339826V2'
Acct-Delay-Time       = '0'
h323-incoming-conf-id= '0CACDDD5 EED3CF4C 8B1E7138 60F09C0B'
h323-incoming-call-id= '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'
h323-gw-address       = '70.68.0.213'
h323-remote-address  = '193.28.87.3'
h323-remote-id        = '193.28.87.3'
xpgk-h323-id          = 'Joe Kotnjek'
xpgk-dst-number-in   = '5555555'
xpgk-dst-number-out  = '5555555'
xpgk-route-retries   = '2'
h323-call-id          = '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'
Exec-Program-Log     = 'porta-billing.pl'
Oct 13 09:09:58: This call belongs to the environment #220 'mvts_test'
Oct 13 09:09:58: Copied account:
16044321[161838,credit,balance=0.00000,limit=100.00000] of customer
VOIP Networks[639,balance=3.00000,limit=none] from '216.231.44.193'
into the current request
Oct 13 09:09:58: PrepareNexecute 'GetActiveLegIdByAcct'
Oct 13 09:09:58: Looking up vendor/connection
Oct 13 09:09:58: Trying to match connection for call
Oct 13 09:09:58: Looking for a connection VoIP/originate
Oct 13 09:09:58: Outgoing VoIP, matching by the remote IP address
'193.28.87.3' (env 220)
Oct 13 09:09:58: Found connection 354 'routes to begemot' to vendor
'World Telecom'
Oct 13 09:09:58: Found vendor/connection
Oct 13 09:09:58: PrepareNexecute 'InsertActiveLeg'
Oct 13 09:09:58: Accounting response
Oct 13 09:09:58: ...Done
```

Accounting Stop from MVTS server

```
Oct 13 09:10:22: Processing request (BE ver1.218.2.16,pid56385):
NAS-IP-Address        = '216.231.44.193'
NAS-Port-Type         = 'Async'
User-Name              = '16044321'
Called-Station-Id     = '5555555'
Acct-Status-Type      = 'Stop'
Service-Type           = 'Login-User'
h323-gw-id             = 'mvts-ua'
h323-call-origin      = 'originate'
h323-call-type        = 'VoIP'
h323-setup-time       = '10:09:57.000 PDT Thu Oct 13 2005'
h323-connect-time     = '10:09:57.000 PDT Thu Oct 13 2005'
h323-disconnect-time  = '10:10:21.000 PDT Thu Oct 13 2005'
h323-disconnect-cause = '10'
h323-voice-quality    = '0'
h323-conf-id          = '0CACDDD5 EED3CF4C 8B1E7138 60F09C0B'
Acct-Session-Id      = '8bc128c6-0-85339826V2'
Acct-Input-Octets     = '9205'
Acct-Output-Octets    = '482'
Acct-Input-Packets    = '890'
Acct-Output-Packets   = '9002'
Acct-Session-Time     = '24'
Acct-Delay-Time       = '0'
h323-incoming-conf-id = '0CACDDD5 EED3CF4C 8B1E7138 60F09C0B'
h323-incoming-call-id = '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'
h323-gw-address       = '70.68.0.213'
xpgk-local-disconnect-cause = '1'
xpgk-src-codec        = 'g7231 '
```

```
xpgk-dst-codec          = 'g711A64k g7231 '
xpgk-initial-inc-local-address = '216.231.44.193'
xpgk-selected-inc-local-address = '216.231.44.193'
xpgk-outgoing-local-address= '216.231.44.193'
xpgk-scd-time          = '0'
h323-remote-address   = '193.28.87.3'
h323-remote-id        = '193.28.87.3'
xpgk-h323-id          = 'Joe Kotnjek'
xpgk-dst-number-in    = '5555555'
xpgk-dst-number-out   = '5555555'
xpgk-route-retries    = '2'
h323-call-id          = '3A3E9C51 3D2C3B45 B98225D5 6BFB0DBA'
Exec-Program-Log      = 'porta-billing.pl'
Oct 13 09:10:22: This call belongs to the environment #220 'mvts_test'
Oct 13 09:10:22: Copied account:
16044321[161838,credit,balance=0.00000,limit=100.00000] of customer
VOIP Networks[639,balance=3.00000,limit=none] from '216.231.44.193'
into the current request
Oct 13 09:10:22: PrepareNexecute 'GetActiveLegIdByAcct'
Oct 13 09:10:22: PrepareNexecute 'UpdateActiveLeg'
Oct 13 09:10:22: End of the outgoing call for logged in account.
Waiting another outgoing call or hang up
Oct 13 09:10:22: Set lifetime with 15s to Thu Oct 13 09:10:37 2005
Oct 13 09:10:22: Looking up vendor/connection
Oct 13 09:10:22: Trying to match connection for call
Oct 13 09:10:22: Looking for a connection VoIP/originate
Oct 13 09:10:22: Outgoing VoIP, matching by the remote IP address
'193.28.87.3' (env 220)
Oct 13 09:10:22: Found connection 354 'routes to begemot' to vendor
'World Telecom'
Oct 13 09:10:22: Found vendor/connection
```

Performing vendor billing operations

```
Oct 13 09:10:22: Charging call
Oct 13 09:10:22: Calculating account's charge by tariff 'Retail Tariff'
Oct 13 09:10:22: Checked 2005-10-13 10:09:57 America/Vancouver against
'hr{0-22} wd{su sa}, hr{23}min{0-54} wd{su sa}, hr{18-7} wd{mo-fr}': 0
Oct 13 09:10:22: Checked 2005-10-13 10:10:21 America/Vancouver against
'hr{0-22} wd{su sa}, hr{23}min{0-54} wd{su sa}, hr{18-7} wd{mo-fr}': 0
Oct 13 09:10:22: Start of call is peak level 0
Oct 13 09:10:22: End of call is peak level 0
Oct 13 09:10:22: PrepareNexecute 'GetPricePerDestination*'
Oct 13 09:10:22: Calculating vendor's charge by tariff 'ABC termination
tariff'
Oct 13 09:10:22: Call to '5555555' with duration 24 seconds will be
charged for 24 seconds and cost is 0 (1x1x0+23x1x0) by rate 451039
Oct 13 09:10:22: Using peak rate, since no off-peak is defined
Oct 13 09:10:22: PrepareNexecute 'GetPricePerDestination*'
Oct 13 09:10:22: Call to '5555555' with duration 24 seconds will be
charged for 24 seconds and cost is 0 (1x1x0+23x1x0) by rate 451040
Oct 13 09:10:22: Charging account for the call
Oct 13 09:10:22: Inserting CDR
Oct 13 09:10:22: PrepareNexecute 'InsertAccountCDR'
Oct 13 09:10:22: Charging account's owner for the call
Oct 13 09:10:22: Charging vendor for the call
Oct 13 09:10:22: Charging vendor 268 'ABC Telecom' 0
Oct 13 09:10:22: Inserting CDR
Oct 13 09:10:22: PrepareNexecute 'InsertVendorCDR'
Oct 13 09:10:22: PrepareNexecute 'UpdateVendorBalance'
Oct 13 09:10:22: Accounting response
Oct 13 09:10:22: ...Done
```