

Porta  Billing[®]



Administrator Guide

Maintenance Release 22

Copyright Notice & Disclaimers

Copyright © 2000-2011 PortaOne, Inc. All rights reserved

PortaBilling@100 Administrator Guide, September 2011
Maintenance Release 22
V1.22.10

Please address your comments and suggestions to: Sales Department,
PortaOne, Inc. Suite #408, 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	5
Hardware and Software Requirements	6
Installation	7
What's New in Maintenance Release 22?	7
Important Upgrade Notes	8
1. System Architecture	9
Overview	10
Components	12
Centralized Configuration Management	14
Updating the System to a New Version	17
PortaBilling®100 Performance	20
PortaBilling®100 Web Interface	25
2. System Concepts	27
Overview	28
Service Types and Services	29
Account and Account Types	32
Products and Accessibility	36
Customers	39
Vendors	45
Account ID-based Billing	46
Nodes	47
Virtual Environments	48
3. Rating and Invoicing	50
Destinations, Rates and Tariffs	51
Billing for Always-on Services	55
Session Billing Parameters	56
Call Billing Parameters	56
Quantity Billing Parameters	64
Overdraft Protection	65
xDR Recalculation	70
Subscriptions	71
Invoicing	78
Processing Taxes	86
Payments	91
4. Voice Services	94
CDR Mediation	95
Call Routing	96
Number Translation	111
Local Number Portability and Other Special Cases	114
Favorite Number Billing	115
Active Call Monitoring	116
IP Device Provisioning and Inventory	116
Using Different Protocols (SIP and H323)	118
5. Converged Services	120

	IPTV.....	121
	Internet Access Service.....	122
6.	Entering Data in PortaBilling® 100.....	125
	Cost/Revenue.....	126
	PortaBilling® 100 Data Model Concept.....	126
	Billing Entities Hierarchy.....	128
	Entering Data.....	129
7.	Management and Monitoring Tools	134
	Templates.....	135
	Date and Time Information.....	136
	Online Web Signup.....	137
	Custom Reports.....	138
	XML API for Data Operations.....	142
	Trouble Ticketing System.....	143
	Billing Server Health Monitoring.....	144
	VoIP Network Performance Statistics.....	146
	Billing Statistics.....	150
8.	How to	156
	Log in to PortaBilling® after installation.....	157
	Locate the h323-conf-id for a call.....	157
	Troubleshoot an incorrectly billed call.....	157
	Make the Periodic Payments tab appear in the customer/account info.....	159
	Make the Custom Fields tab appear in the customer/account info.....	159
	Make a custom report from PortaBilling®.....	160
	Use ODBC to connect to PortaBilling®.....	161
	Force PortaBilling® to disconnect after a customer calls over his credit limit.....	166
	Create accounts to be used for SIP services.....	166
	Integrate PB logins to your website.....	166
	Add your company name and logo to every PortaBilling® web page.....	167
	Add a reseller's logo to his customers' web pages.....	167
	Frequently Asked Questions (FAQ).....	169
	Glossary / List of Abbreviations.....	170

Preface

This document provides PortaBilling®100 users with the most common examples and guidelines for setting up a VoIP network or enhancing a network with services such as IP TV. The last section of the document answers the most frequent questions users ask after running PortaBilling®100 for the first time.

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/support/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



The **exclamation mark** draws your attention to important information or actions.

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



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



Tips provide information that might help you solve a problem.

Trademarks and Copyrights

PortaBilling®, PortaSIP®, PortaUM® and PortaSwitch® are registered trademarks of PortaOne, Inc.

Hardware and Software Requirements

Server System Recommendations

- Three (3) UNIX Servers for the PortaBilling®100 Standard: the real-time billing engine (RADIUS, master copy of the database), the Web Interface (Web server, XML API services, periodic tasks, invoicing, slave copy of the database), and configuration management (this server is also used for collecting and processing log information). For PortaBilling®100 Advanced you would need five (5) servers.
- A minimum of 120 GB of available disk space on each billing server. On the average, 200,000 CDRs take up about 1 GB of disk space (this includes database files, indexes and binary logs, raw Radius detail files, billing engine log files and other related information), plus you need to reserve an amount of free space roughly equal to the projected database size for performing operations such as backup. RAID is recommended in order to improve performance and reliability.
The configuration server requires about 500GB of free disk space if you are using PortaSwitch® (where log files from PortaSIP® and PortaUM® nodes need to be processed), and 250GB of free disk space if you are using the standalone PortaBilling.
- An i386 processor (Xeon, Opteron) with 64bit support. Additional processor speed is needed for networks with a high call volume.
- At least 4 GB of RAM, 8 GB recommended.
- At least one USB port.

For additional details and configuration advice, see the *Hardware Recommendations* topic on our website:

<http://www.portaone.com/support/faq/hardware-requirements/hardware-requirements/>

For information about whether particular hardware is supported by Oracle Enterprise Linux from the JumpStart Installation DVD, consult the related document on the Oracle or RedHat website:

<https://hardware.redhat.com/>

Client System Recommendations

- OS: Windows XP, Vista or 7, UNIX or Mac OS X
- Browser: Internet Explorer 7.0, FireFox 3.x with JavaScript enabled.
- Spreadsheet processor (MS Excel)
- Display settings:
 - Minimum screen resolution: 1024 x 768
 - Color palette: 16 bit color (minimum)

Installation

A jumpstart installation DVD is provided for all PortaOne products. This DVD contains installation media for Oracle Enterprise Linux (64-bit version), supplementary packages necessary for convenient system administration and maintenance, and all required software packages. After the installation is complete you will assign roles (e.g. RADIUS, web interface, etc.) to individual servers using the configuration server tool – this will automatically enable the required components of PortaBilling®100 software on each server.

PortaBilling® installation and configuration are automated and integrated within the main installation process. This allows you to install a completely functional PortaBilling® environment (two servers) from scratch in less than one hour!

For detailed installation instructions, please refer to the [PortaSwitch Installation Guide](#).

What's New in Maintenance Release 22?

This release includes several new features and improvements:

- **Payback (reverse) charges** – ability to credit customer's account for certain types of activity (e.g. receiving incoming calls).
- **Subtotals in custom reports** – it is now possible to include subtotals for individual groups of data in custom reports.
- **Routing calls between multiple nodes** – if your network contains multiple nodes, you may choose to designate one of the nodes as the “master” routing entry. The other nodes will then send off-net calls to that node, which will do the actual routing to the carriers.
- **Collection threshold** – invoices with very low amounts due will be ignored by the collection process, so notifications about the unpaid or overdue invoice will not be sent and no suspension will result from unpaid invoices.
- **Support for WiMAX services** – PortaBilling® RADIUS server now supports WiMAX RADIUS attributes and can be used as a AAA server in conjunction with WiMAX ASN.
- **Extended overdraft protection support for forwarded calls (since MR22-3)** – The same approach as that used for dynamic re-authorization of outgoing calls can now be applied to calls that are forwarded using call forwarding / follow-me. This guarantees that prepaid or postpaid accounts will not consume more than the

available funds allow even if multiple calls are being forwarded simultaneously.

Important Upgrade Notes

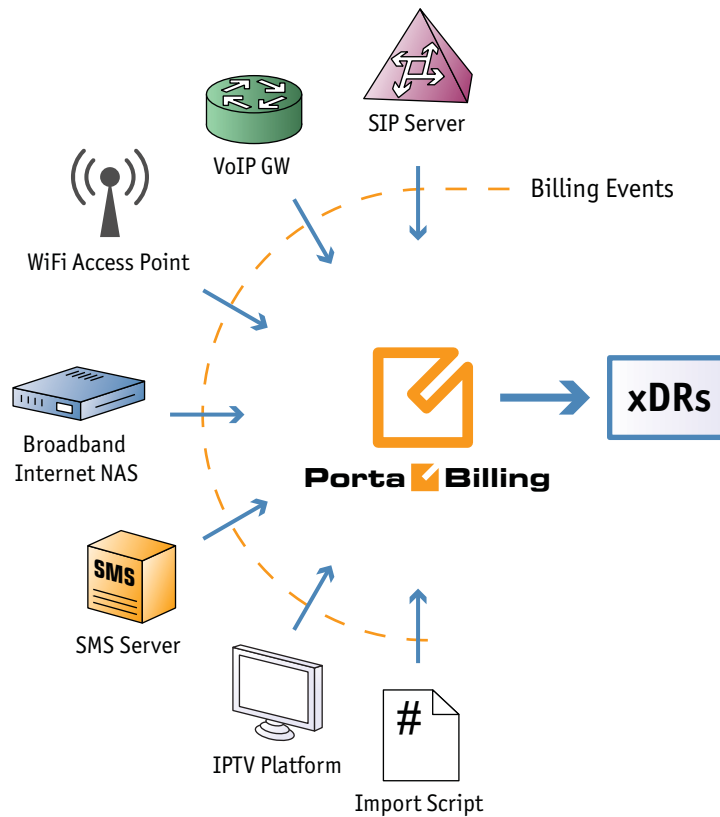
We try to make the process of upgrading as easy as possible, and to keep our releases backward compatible. Here are just a few things you should remember when upgrading:

- Since the FirePay payment processor service is no longer available, it has been removed from the list of available payment types on the **Payments** page.
- If you are using the **Follow-me** service or charging customers for incoming calls, then we recommend upgrading your system to MR23 which has improved overdraft protection for these services. Alternatively, you may want to use **Follow-me** only for post paid services where strict overdraft protection is not vital.

1 . System Architecture

Overview

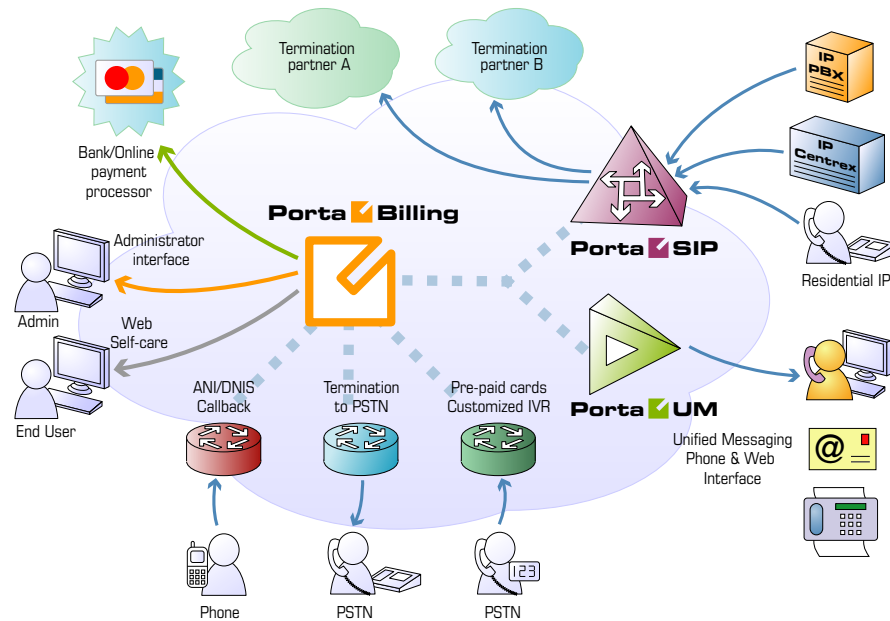
PortaBilling®100 is a carrier-grade converged billing/provisioning system for the IMS (IP Multimedia Subsystem) environment. It communicates with elements of your network (such as VoIP gateways), provides these nodes with authentication or authorization (determining whether a customer should be admitted and provided with a service), and gathers billing events, i.e. data about services rendered to your customers. Based on this information, it performs rating for the services, creates transaction records (also called xDRs - eXtensible Detail Records¹), and modifies customers' balances accordingly.



All this happens in real time, so the billing data is updated as soon as a session is completed (e.g. the customer hangs up his phone, or an SMS message has been sent). PortaBilling®100 provides a unified platform for multiple services, which allows you to use it to charge clients for their voice calls, messages, and data transfer, thus effectively deploying triple-play on your network.

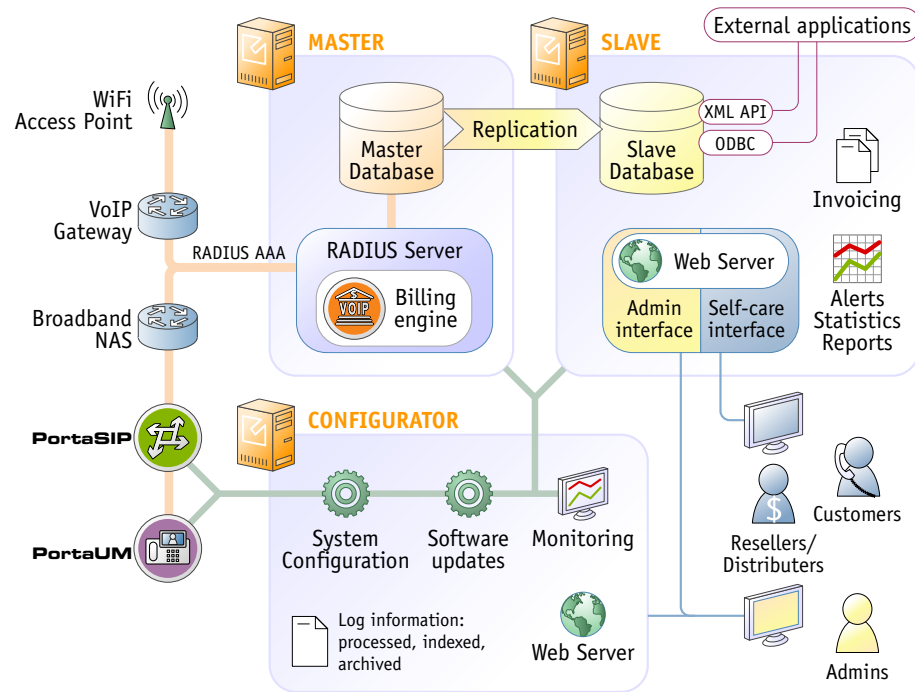
¹ For xDRs for telephony services, the previous term CDRs (Call Details Records) is often used.

PortaBilling®100 will act as the nerve center of your network. After you have entered information about your services, rates, customers and so on via the web interface, PortaBilling®100 will communicate in real time with elements of your network to supply information regarding which customers the service should be provided to (and which not), as well as exactly how it should be provided. Customers whose balance has run out will be disconnected immediately after exceeding the maximum session duration and (since billing happens in real time) funds will be withdrawn from their account and service denied if they make another attempt to use the service.



As shown in the diagram above, multiple types of network equipment can work in conjunction with PortaBilling®. The easiest and best method of integration is via the RADIUS protocol. However, if a certain type of switch or server does not support RADIUS, integration can be done by other means, for instance XML API.

Components



PortaBilling® consists of three main logical components: the master or RADIUS server (left), the slave/admin/web server (right), and the configuration/log collector server (bottom). Typically, the master and slave servers are installed on two separate physical computers so as to provide data redundancy and load sharing. Although an “all in one” combination (where both servers run on a single physical computer) is also possible and supported, this is recommended only for evaluation or demo systems.

The configuration server must always reside on a separate server. It alters the software configuration and version of the other servers, and so must remain unaffected by the changes thus made. This allows a rollback to the previous stable version of the configuration or software release.

Master Server

The master server hosts the master database and radius daemon. Radius has a billing engine embedded into it, and so after it receives a request from the gateway it passes processing on to the billing engine and sends back a response.

Billing engine

The billing engine provides:

- Authentication – It tells the node whether the subscriber (identified by phone number, PIN, IP, or the like) is allowed to

use a specific service (e.g. voice calls or wireless Internet access), and returns attributes such as current balance.

- Authorization – It tells the node whether the subscriber (identified by phone number, PIN, IP, or the like) is allowed to initiate a session with specific parameters within a service (e.g. calling a particular phone number), and returns session attributes such as the maximum allowed session duration or the allowed amount of bandwidth.
- Interim (also called keep-alive) accounting processing – This is used for services like Internet access, when nodes constantly update the billing about usage for the current session.
- Accounting processing – Based on information from the gateways, it bills the session and writes transaction records to the database.

The billing engine also keeps a cache of current sessions in its memory, allowing PortaBilling® to:

- combine information from different network nodes (NASes, voice gateways, proxies, etc.) in order to obtain complete information about a session;
- reuse information from earlier steps so as to enhance performance.

Slave Server

The slave server hosts the following:

- Web interface, consisting of:
 - Admin interface
 - Account self-care
 - Customer self-care
 - Reseller's helpdesk (self-care; customer care)
 - Vendor self-care
 - Representative self-care
 - Online web signup interface
- Slave database
- Scripts for calculating various statistics
- Optional IVR (TFTP) server
- XML API (SOAP) interface for integration with external applications

Configuration Server

This server stores the information about all the other servers in the installation plus service configuration details about each of them. See more details about the configuration management framework in the *Centralized Configuration Management* chapter below.

The configuration server monitors the status of all the other servers. Data such as available disk space, load, or service status can be viewed via the web interface. If a problem is detected, the PortaOne support team will be notified immediately.

The configuration server also collects log data, such as SIP communication logs from PortaSIP, or billing engine logs from billing servers. These are then stored, processed and indexed, so that all entries relating to a particular session can be retrieved instantly. This reduces the amount of disk space needed on the production servers, as well as the load on them.

Log data for the most recent time period is kept in the original format. Older data is archived to save disk space, while log files which are outdated are erased.

Centralized Configuration Management

In order to efficiently maintain large PortaBilling® and PortaSwitch® installations (which may involve 10 or more servers), it is essential to have a unified interface for managing all the configuration data. Tasks such as IP address changes, relocating services to different physical servers, or simply changing an option that affects functionality can then be performed quickly and easily, with a minimal chance of error.

Configuration server carries out exactly this task, providing an interface for the administrator to view the current configuration, create a new configuration and correctly apply it to all servers, or rollback to an old configuration if a problem has been detected. Another important role of the configuration server is that it stores “images” of different versions of the software. Each image is the actual content (in a binary format) of a specific version of the software code (e.g. Maintenance Release 21, build 3). When a specific image is loaded, the server will operate under the corresponding software release.

Concepts

There are several important concepts involved in the configuration management framework. Configuration management is designed to work in the same way whether it is controlling just a single PortaBilling® installation (two servers) or a full PortaSwitch® Procinctus (twelve servers). In the rest of this section, we will use some examples related to managing the full PortaSwitch® configuration, so as to better illustrate the capabilities of the configuration framework.

Server

A server is an atomic element of processing capacity. It is either a single physical server, or a separate virtual machine, if virtualization is used. In other words, it is basically a host on which PortaSwitch® software can be installed and operated. A server has attributes such as a number of available CPUs, disk space, and so on.

Role

An operational PortaSwitch® system performs many hundreds of different functions, from sending a “You’ve got voicemail” notification to an IP phone to providing a website where new customers can sign up for a service. To control each of these functions individually would be a tedious task, and so closely-related functions are grouped into sets. Such a set of tasks, e.g. “Web interface for customer self-care” or “Invoice generation”, is called a role.

The main purpose of roles is to be assigned to specific servers where these tasks will be actually executed. As in real life, we group a set of tasks (making a shopping list, driving to the store, putting the items we need into a cart, paying, driving home) into a role called “doing the shopping”, and then assign this role to someone in the family. Some roles can be assigned simultaneously to multiple servers (e.g. several servers are concurrently running PortaSIP instances), while others (e.g. “Master database”) can only be assigned to one server at any given moment. Also, such clear separation of roles and the entities they are assigned to allows them to be re-assigned easily, just as you can re-assign the role of grocery shopping to a different family member.

Instance

An instance is a combination of the software code, configuration data, and running processes that provide an actual service. For example, the role PortaBilling® RADIUS assigned to server ABC creates a service instance, so that RADIUS AAA requests from the network can be received and processed. So while the role could be compared to a building blueprint (a formal description of what should be done), and the server to a vacant building site (some land where various things could be done), the instance would be the building actually built.

Instances are especially important when the server is executing the same role several times simultaneously, e.g. a single physical server ABC running a PortaSIP role, but with 3 separate virtual PortaSIP environments on that server. Thus each of these virtual environments is an instance.

Option

An option is a configuration parameter which alters the system's functionality. Some examples of options would be "Which IP address should the PortaSIP service run on", "When should statistics generation be done", or "Should the previous balance be included in the invoice's amount due". The value of an option could be a on/off switch, one of the choices made via a select menu, or a text string.

For convenience in administration, a default value is provided for most of the options, so that you do not have to supply a value for every single option in order to make the system work.

Configuration tree

The full system configuration includes hundreds of different options, so it would certainly be inconvenient to work with them as a single list. Thus options are grouped together in a tree-like structure.

On the top level of the tree are the global options, i.e. those that have an effect on all components of the system.

Below the global level is the role level. Each role is presented as a separate node in the tree, so in order to change the option "When should statistics generation be done" (which is related to the PortaBilling® invoicing role), you would first open the "Invoicing" node.

The next level is the environment level. This provides control for options specific to a particular virtual environment. Finally, there is the instance level, which covers options related to a specific instance (e.g. start accounting is enabled on the PortaSIP instance with IP address 1.2.3.4).

When the value for an option is defined on some level, it is automatically propagated to all levels below; e.g. if the "start accounting" option is activated on the global level, start accounting will be active by default for all PortaSIP instances in all environments. It is possible to override options on lower levels; e.g. in our previous example, "start accounting" can be selectively disabled for a specific virtual environment or a particular instance.

Since normally there are many individual options available on each level, it would be inconvenient to work with them if they were organized into a single long list. Thus options are split into groups, with each group containing a small set of options related to a single software module or feature.

Multiple configuration trees

When a configuration is saved, this stores all the options in it – so every stored configuration contains a complete set of data for operating all PortaSwitch components. In order to preserve system integrity it is not possible to directly alter the active configuration (the one currently applied to the servers).

In the case of changes not producing the desired result, it is always possible to roll the system back to its original, stable state.

The process of changing the system configuration is thus divided into several steps:

- Clone the current configuration tree into a new one;
- Make the required changes;
- Now apply this configuration to the system so that it becomes the **active** one.

Applying the configuration

Every server in PortaSwitch® runs a configuration update agent, which follows commands from the configuration server. When a configuration update is received, the agent updates the local files, restarts the processes and does everything else required to put the changes into effect.

Updating the System to a New Version

Updating your system to a new release (whether it is your personal WiFi router or a powerful PortaSwitch® server, serving tens of thousands of customers) is always a challenging task. You ask yourself questions such as: How long will it take? Will updates be applied correctly to all system components involved? What happens if something goes wrong – can I get my system back to the operational state it is in now, etc.?

PortaSwitch® utilizes an innovative system of maintaining and modifying the software code on each server, allowing you to properly address all of the concerns expressed above and ensure that you are able to:

- Migrate quickly to a new maintenance release, without any problems on the way and obtain the system that operates 100% according to how it is intended to operate.
- In case there is something wrong with the functionality of the new release (e.g. you just realized that in order to properly use the new feature you need to train all of your staff, and this would take several days) you can safely rollback to exactly the same version of the software you were using prior to the update.

Most of the software systems currently used throughout the world contain a single copy of the “current” software so that when an update is done – some parts of it would be replaced with updated versions. This brings up a significant risk of incompatibility between the updated components – and the ones that remain from before the update may render the whole system unstable.

An alternative approach – when the entire code image is replaced with a new version (this is how you update the firmware in your router, for instance), poses the risk that if something goes wrong during the update process – the system ends up without any operable software and becomes totally unusable (my guess is that you probably heard about the iPhones which “brick” after an update?).

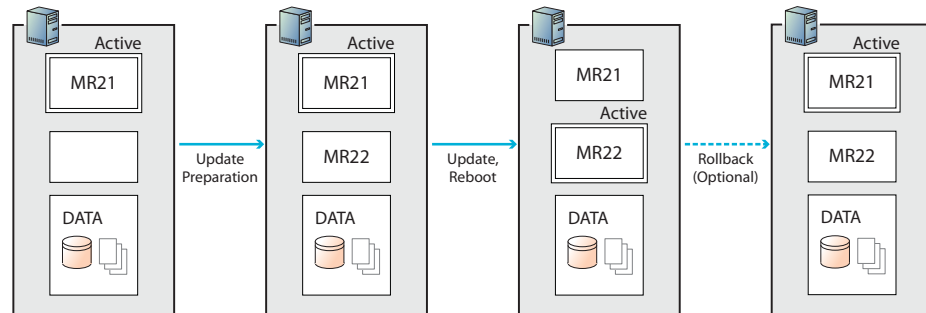
This is why PortaSwitch® utilizes a dual software version management system that has none of the weaknesses described above.

The disk subsystem on each PortaSwitch® server contains three (3) separate partitions. One of them is used to store the actual application data, i.e., database files, logs and .CSV files with statistics of the customer’s activity, etc. Two (2) other partitions are equal in size and each of them can contain the full set of the software “code” required to operate the server – operating system, third-party libraries and modules (e.g. Apache) and the actual code for a specific application, e.g. PortaSIP®. At any given moment one of these partitions is considered active: this means that upon startup, the server uses this particular partition to boot up and the application code, located within it, is used to operate the service. When the system is being prepared for an update to a new release, the other partition is cleared and the new version of the code is installed there. This is done while the system is still operating under the current version of the software, without any service interruption. Now the server has all the required data to operate with the new release – moreover, since the new release is installed as a set of binary packages, one can be sure that this is exactly the same code (the same version of operating system, the same version of kernel and the same bytes in every single utility or file!) that was used in PortaOne’s labs during the testing period, that was deployed on staging systems during the field testing and that is currently being used by other PortaOne customers worldwide.

After that, the configuration agent updates the “local” files (e.g. “/etc/hosts”) based on the system’s configuration stored in the configuration server: e.g. what IP address each service is working on and which application-specific features are on and/or off, etc.

Finally, at the specific time the new partition is marked as “active” the server is restarted using the new version of the code (these tasks are done automatically by the update agent, controlled by the configuration server). The potential downtime is just a few minutes – the time required to

complete the restart. Nothing is changed in the “old” partition though – so if a rollback is required, it only requires a reboot from that partition and the server is back to the old, “stable” release.



After some time when you wish to update to an even newer release, this partition is wiped clean and the new version of the code is loaded into the recently emptied location. Then the process described above repeats.

The same process is used to update to a new maintenance release or to a newer software build within the current release.

Updating the Application Data

If all PortaSwitch® applications were “stateless”, in other words, if they were only doing some calculations based on the current input from the user and some pre-programmed rules – this chapter would end with the previous paragraph.

Actually, most of the applications in the real world, and PortaSwitch® is one of them, rely on a large set of previously accumulated data to make decisions about how the service should be provided. For instance, if the customer has already made calls to the US & Canada for a total of 101 minutes during this billing period, and his plan only allows for 100 free minutes – a call made right now would be charged at \$0.05/min rate.

All the data accumulated by the old software release are available to the new one after the upgrade to ensure the system’s proper operation – and this involves changing the data files, database structures and data to accommodate the new release.

So the update process includes two extra steps:

- Non-blocking data modifications are done as part of the “preparation” process, when the new release is already installed to the new partition, but before it becomes active. These modifications include adding new tables, inserting new records, etc. – basically anything that could be done while the system continues to operate with the old release.

- Blocking data modifications are those that may affect the system's performance while they are being carried out. For example, adding a column to a table in MySQL would stop all other queries to that table from being executed until the operation is complete (another advantage, offered by PortaBilling® Oracularius is that there are almost no "blocking" operations there). Blocking updates are done during off-peak periods. Needless to say, the PortaOne team tries to reduce the amount of blocking data structure modifications and the amount of time required for applying them.

The process described above allows the data modifications to be performed while the system still operates using the current release. There is one important consideration, though: during that time, the "older" version of the release operates with the "newer" version of the data. (The same situation would happen if you were to rollback to the older release).

The PortaOne team has a development and testing process specifically aimed to make this possible, but we can only guarantee this interoperability for the adjoining releases. In other words, the system operating with MR21 will operate normally while the data is being updated to MR22 (or it is possible to rollback to MR21 after the migration to MR22 has been done). It is not possible to provide this transparency when the distance between releases is too great (e.g. MR21 will most likely *not* operate properly if the data has already been updated to the MR24 format). Thus, the preferred method of updating that provides unlimited time for update preparation and allows for fallback to the previously used version is to go step by step: e.g. from MR21 to MR22, then from MR22 to MR23 and finally from MR23 to MR24, etc. (If required, it is still possible to do a MR21 to MR24 update in one go, of course – but in this case there is no possibility of performing a rollback).

PortaBilling® 100 Performance

When buying a new car, you are obviously interested in performance parameters, such as "how fast can it go?". The maximum speed you can achieve is, of course, dependent on the clearly defined technical parameters of your car, such as engine horsepower or the car body's total weight. So, in general, the more powerful car you buy, the faster you will be able to move. However, the actual speed of the car at any given moment depends on many other criteria: how many passengers are in the car, how much weight there is in the trunk, what the road conditions are (a paved highway vs. a gravel road), and so on.

The situation with billing systems is quite similar. The questions asked are: "How many customers can I have?", "How many concurrent calls can it handle?", or "How many messages or call minutes per month can I

transport on my network?”. The actual numbers depend not only on the PortaBilling® functionality and your hardware’s performance, but also on your billing configuration, traffic patterns, and the types of services provided. Let’s take a look at the details.

The only substantial performance criterion for PortaBilling®100 is the speed with which it can process requests from your network. PortaBilling®100 can process up to 100 RADIUS requests per second. What exactly does that mean? Every call attempt or completed call on your network generates several RADIUS requests, with the total number of requests and their type depending on the kind of service provided. For prepaid VoIP services, which involve call routing, this translates to about 20 call initiations per second, while for postpaid VoIP services there are about 35 call initiations per second. This means that 20-35 users can start a new phone call on your network each second. The exact figures will depend on the complexity of your billing configuration (e.g. how many rate plans, individual rates in a rate plan, or carriers you have). Since it cannot be known beforehand what combination of different services you will be providing on your network, it only makes sense to refer to an **estimated** PortaBilling®100 processing speed, which has been calculated at **26 session initiations per second**.

How many simultaneous (concurrent) calls does that translate to?

This primarily depends on the average call duration and call success rate. Assuming an aggregated call processing speed of 26 call attempts per second, an average call duration of 5 minutes, and a call success rate of 50% (the industry norms), it means that 50% of the 26 call attempts per second will succeed. So 13 calls will be connected, while the same amount of previously connected calls will be disconnected. Since the average call duration is 300 seconds, this means that at all times approximately $13 * 300 = 3,900$ calls will be in a “connected” state. If your ASR changes, it will have an immediate impact on the number of concurrent calls, e.g.:

- ASR 25%: $(26 * 25%) * 300 = 8.5 * 300 = 1,950$ concurrent calls
- ASR 60%: $(26 * 60%) * 300 = 24 * 300 = 4,680$ concurrent calls

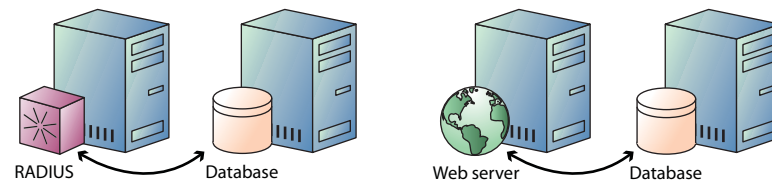
How many minutes of monthly traffic can PortaBilling®100 support?

Again, this depends mainly on your traffic patterns (the ratio between the duration of your peak and off-peak hours and the differing amount of calls in peak and off-peak hours) and your call success rate. We can estimate this based on typical traffic patterns (as indicated below) with daily peak hours, medium load hours and off-peak hours. We should also bear in mind that your network must be able to handle all kinds of traffic – off-peak hours, peak hours, and extraordinary traffic splashes (e.g. New Year’s Eve). Thus the total performance capacity must be greater than the

maximum anticipated BHCA (busy hours call attempts). On average, extreme traffic peaks (such as Christmas, New Year's, etc.) will have 2 to 3 times more call attempts than your normal peak hours, so approximately $26 / 3 = 9$ call attempts per second can be expected during your peak hours. Using the same "industry norm" figures as above (50% ASR, 5 mins ALOC), PortaBilling®100 can process $(9 * 50\%) * 5 * 3600 = 81,000$ minutes during your peak hours. We can estimate that your total amount of daily traffic will be 8 times the amount of traffic during one peak hour, i.e. $100,000 * 8 = 648,000$ minutes per day. Then, multiplying the daily amount by the number of "business" days in a month, we get $648,000 * 20 = \mathbf{13,000,000}$ minutes. This is why (to be on the safe side and allow for some reserve capacity) PortaBilling®100 is poised to handle monthly traffic of slightly over **10 million minutes**.

Dedicated Database Server

While the default installation of the standard version of PortaBilling® puts the database on the same server as the applications (so the master database runs on the same server as RADIUS, and the slave database runs on the same server as the web interface), in PortaBilling® Advanced the application servers are separated from the database for enhanced performance. The extra processing power and increased database in-memory cache on the dedicated server dramatically reduce the time required to execute many database queries.



PortaBilling®100 SMP

For increased performance, you can use PortaBilling®100 SMP (Symmetric Multi Processing). This is a specially optimized version of the PortaBilling®100 billing engine using multiple processors in your server to process several requests in parallel, thus boosting performance. PortaBilling®100 SMP technology is included in PortaBilling®100 Advanced, and since PortaBilling®100 Advanced uses the same database layout as the standard PortaBilling®100, all the administration tools (e.g. the web interface) are exactly the same. A standard PortaBilling®100 installation may be easily upgraded to PortaBilling®100 Advanced at any time.

Performance

Assuming that the disk subsystem in your servers provides adequate data access speed, upgrading from a single-CPU server running PortaBilling®100 to a dual-CPU server running PortaBilling®100 Advanced will yield a 2-3 times improvement in performance and will allow you to process up to **60** call attempts per second.

PortaBilling®100 Cluster

In addition to the ability to run several RADIUS processes in parallel on the same machine, you can also utilize multiple physical servers to process RADIUS requests. This provides extra performance, since incoming requests are distributed for processing among all the available servers. It also offers you improved reliability, because if one of the servers is down due to hardware failure, the remaining servers will continue operating. RADIUS cluster technology is included in the PortaBilling® Oracularius and PortaSwitch® Procinctus products.

Technical details

Each server in the cluster runs two types of processes: dispatcher and processing engine. The dispatcher receives a RADIUS request from the network and sends it to one of the available processing engines. The actual business logic (e.g. checking whether it is a valid account/customer, calculating allowed session duration, or producing CDRs) is done by the processing engine. Normally there is a single dispatcher and multiple processing engines on each physical server.

One of the dispatchers is always active, while all the dispatchers constantly communicate and thus know which nodes in the cluster are alive. Should the server which runs the active dispatcher go down due to hardware failure, a new active dispatcher will be selected. The active dispatcher places requests into queues for the processing engines, based on their availability and current load. When the processing of a request is finished (meaning that xDRs are inserted in the database, balances are updated, etc.), the dispatcher receives confirmation from the processing engine and relays the response to the network node (this may be a simple accounting confirmation, or could include attributes such as “maximum authorized time”). If one of the servers in the cluster goes down, all requests queued for processing engines on that server are processed by the remaining servers.

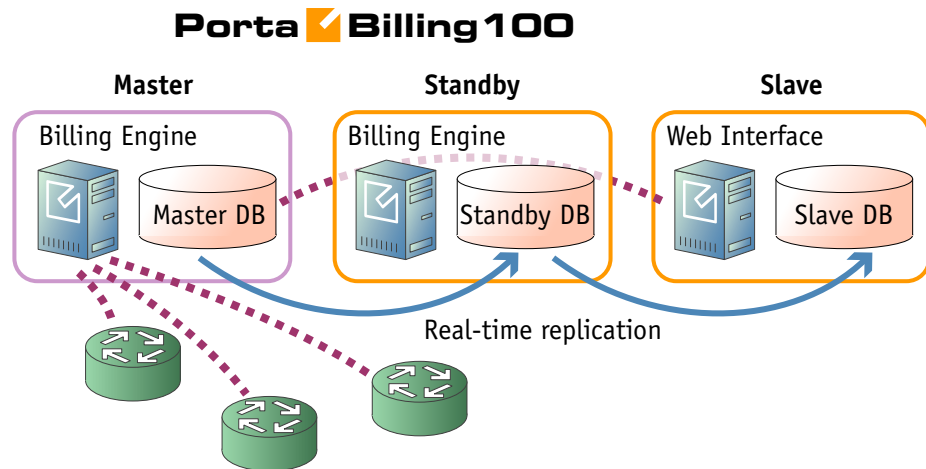
Performance

Since the load is distributed among all servers in the cluster, the total performance capacity grows linearly with adding more servers into the cluster. So if we add to a PortaBilling Advanced system, capable of **60** call attempts per second, another identical server and connect these two into a

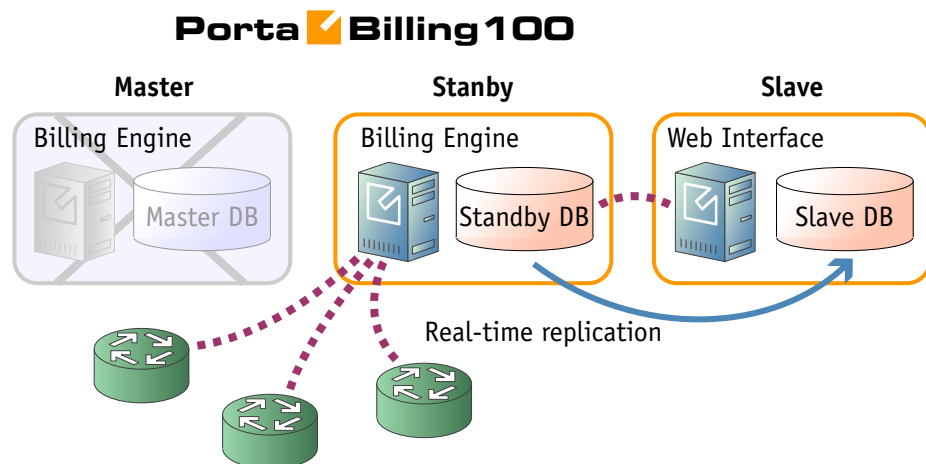
cluster – the performance of this two node cluster will be about **120** call attempts per second and by adding a third node it can be increased to **180** call attempts per second.

Using Standby Database for Improved Reliability

For quick disaster recovery in case your master billing server goes down (e.g. due to motherboard failure), PortaBilling®100 can be equipped with a standby server. This server is identical to the master server, with the same software installed. The only difference is that during normal system operations the database runs in slave mode (with real-time replication), and so is fully synchronized with the primary master database.



When the primary master is unavailable, the standby server can be switched to the role of primary master and assume control of your VoIP network.

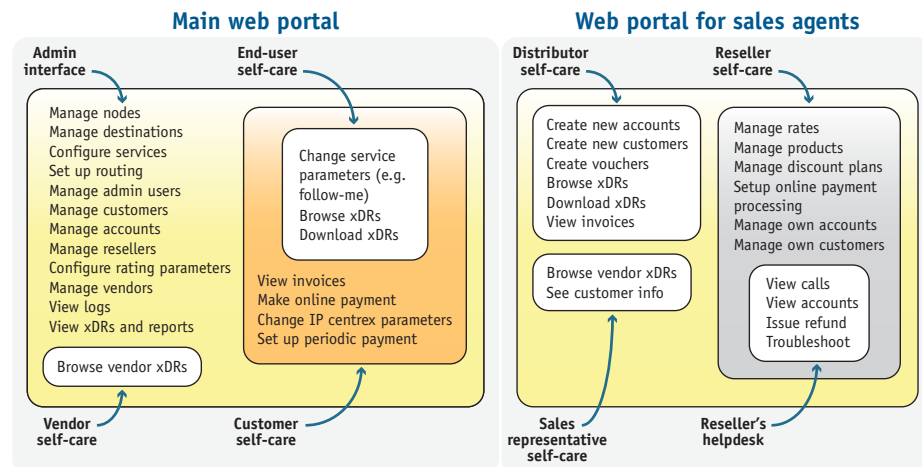


The standby server may be installed in the same co-location center (even in the same rack) as your primary master server. This provides easy administration and, in case of a switch from standby to master, the standby server is able to simply take over the previous master's IP address. In such a case, the change is transparent for all elements of your VoIP network.

However, should your co-location provider be having a problem with network connectivity, both servers will become inaccessible to your VoIP network installed elsewhere. This can be avoided by installing the standby server in a different physical location, connected to a different network (and preferably to a different ISP). In that case, even if the whole hosting center including your main server is down, you can still continue network operations via the standby server, since this will not be affected by the outage.

PortaBilling® 100 Web Interface

Different operations are available for different types of users who access the system (also, the abilities of an individual user can be adjusted via his access level settings). In general, administrators (PortaBilling® users) can potentially perform all possible configuration tasks, while others are limited to their own entity and related objects only. For instance, a reseller can access any of his sub-customers or their accounts, while a sub-customer can only access information about himself and his accounts on his self-care pages. The self-care interface imposes multiple security-related limitations on a user's activity, e.g. a sub-customer can view his own balance on the self-care pages, but cannot modify it.



For improved security, the PortaBilling®100 web interface is divided into several separate areas, each serving its own purpose. Thus a login name and password which allow a user to access the customer self-care interface cannot be used to log in to the admin interface; the login name and

password for the account's (end-user's) self-care interface are not applicable to customer self-care; and so forth.

2. System Concepts

Overview

Converged Billing

PortaBilling® supports multiple services and service types. This means that as different types of services (e.g. voice calls, WiFi connectivity and messaging) are provided to your users, PortaBilling® collects data about all of them, processing and rating it according to the billing configuration. It then provides your customers with a consolidated bill (and your administrators with a unified customer management interface).

Billing Events

The main unit of billing information is a billing event – a notification that a service has been provided to a customer in the outside world, and that this customer should thus be charged for that service.

For many services (such as SMS messaging), a single billing event is represented by a single notification message to billing, while for others information about one event (e.g. a completed phone call) is split into multiple notifications from different network elements.

4-tier Billing Scheme

Four separate entities will be billed for every event registered in the system:

- Account (identified entity using the service)
- Customer (an account owner)
- Reseller (optional)
- Vendors (service providers, e.g. your termination partners or service providers for incoming telephony lines, both local and toll-free)

This allows you to:

- Bill the end user (the owner of a prepaid calling card, phone number or office VoIP gateway)
- Bill the customer to whom the end user belongs
- Bill the reseller who owns that customer (if applicable)
- Calculate your expenses for termination of this call to the vendor

xDRs

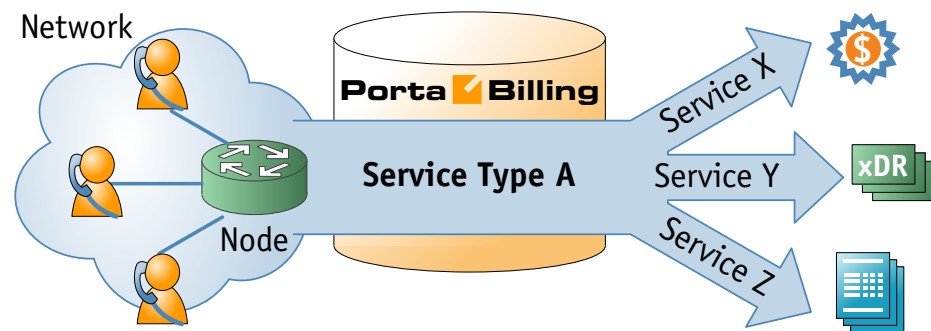
CDRs (Call Detail Records) were the real foundation of any legacy billing system, which had to receive, store and analyze them. Today, when most service providers are expanding their service portfolio, the CDR concept is either becoming inapplicable to some of their services (e.g. selling pizza over the Internet), or it no longer fits their business model (e.g. providing bulk free calling minutes). PortaBilling®100 supports this by introducing the concept of xDRs – eXtensible Detail Records, which can store information about various types of billing events. CDRs are, therefore, only a subset of all possible xDR instances. The same applies to EDR (Event Detail Records), data which describe a single event such as an SMS message; they too are just a subset of xDR data, and can thus be easily processed by PortaBilling®100.

Service Types and Services

PortaBilling® enables you to offer multiple services to your customers, so you can broaden your range of services while providing clear transaction and invoice information to your subscribers and administrators.

There are two entities involved in the billing configuration:

- Service types
- Services



A **service type** is a description of the physical service provided to an end-user. A service type primarily defines the way in which the billing engine interacts with network equipment to ensure that the service is authorized, provided and rated correctly. VoIP calls and dialup or WiFi Internet access are examples of service types. The list of available service types is determined by the billing engine capabilities in a given release, as new service types appear when proper support is added to the billing engine internal modules.

A **service** is a description of the user's activities from a business perspective. Services are the foundation of your billing configuration, and new services may be defined according to your business model. This permits flexible branding of your services to customers and opens the

possibility of bundling various services into one product. You can choose names and billing parameters for new services as desired. For instance, the same physical service, such as Internet access, can be sold to one category of users as “High-speed Internet”, and to another category of users at a different rate as “Broadband connectivity”. Moreover, the **same** physical service can be sold to the **same** customer as multiple entities: for example, the use of WiFi in the customer’s office and in an airport are treated as two separate services with a different price. A breakdown of charges per service is provided on the invoice and in the xDR browser.

Rating base

When information is received about the use of a particular service type, the billing engine must process it and decide which attribute or set of attributes are to be used later to calculate the charged amount. This may be quite different depending on the service type: session duration, amount of transferred bytes, counter values, and so on. The logic determining which attributes are used for billing is called the rating base. For instance, the rating base for the **Internet Access** service (when charging for a data transfer) will define what exactly constitutes the “traffic” which the customer is being charged for: the amount of downloaded data, the amount of uploaded data, or both.

The most commonly used rating bases are:

- **session-time** – These include phone calls, Internet access sessions and other services where the basis for calculating charges is the duration of time in which a customer used the service.
- **quantity** – These services include data transfer, text messaging, pay-per-view movies, or pizza delivery orders. The duration of the service session is either unsuitable or irrelevant for billing purposes. Rating is done based on the type of service element used (e.g. if the movie purchased was a “New Release” or “Classic”) and the quantity (did you order two pizzas, or just one?).

Each service type supports at least one rating base. Potentially, however, it can have more than one, e.g. the **Internet Access** service can be charged based on either session time or the amount of data transferred.

Every new service has a name and several important parameters:

- The service type it uses (see below for a description of available service types).
- The rating base, defining which set of attributes will be used to calculate the charged amount.
- A base unit name; base units are those units in which the amount of the service “used” is reported to billing (e.g. information about session duration is usually given in seconds, while information about the amount of data transferred is given in bytes). You

cannot change the type of base units, since this is typically defined by the configuration and capabilities of your network equipment. you can, however, give it a name which is easy to understand, or even put it in your user's local language.

- Billing units are used for all functions in PortaBilling®. These are the units in which your pricing is specified, in which customers see the amounts on their invoices, and so on. For instance, the billing units for most telephony calls are minutes, while in the case of data transfer services it could be kilobytes (for a GPRS data service) and megabytes or even gigabytes (for a broadband Internet connection service).
- Obviously, you need to define how your billing units relate to your base units. PortaBilling® gives you full flexibility to decide whether 1 kilobyte will equal 1,000 or 1,024 bytes for billing purposes. Rating schemes for other services may be defined with the same degree of flexibility.

Measurement, base and billing units

Measurement units are the units in which the billing engine receives information from the network. Obviously, the billing engine needs this information in order to interpret amounts correctly. However, a PortaBilling administrator rarely comes into contact with these units, since for the purposes of adjusting the rating configuration he would use the following:

- Base unit - The smallest elements that can be rated. This can be thought of as the “rounding threshold”. For example, if a customer has downloaded 758 bytes, but the base unit chosen is a kilobyte, for all further rating calculations he is considered to have downloaded 1 kilobyte.
- Billing unit – The unit in which it is most convenient to express the amount of services used. The price of the service will also be defined using this unit.

The base unit must be equal to or greater than the measurement unit, and the billing unit must be equal to or greater than the base unit. Thus while in some cases the units will be equal (e.g. call duration is reported in seconds and the base unit for the voice calls service is also one second), in other cases you can scale units to make them more convenient for your administrators and customers. Broadband Internet services are one good example. Although network equipment always reports the amount of transferred data in bytes, it would certainly be very confusing to report the total downloaded data as “234564474 bytes”, or inform customers that “your current rate is \$0.0000000014 per byte”. They will be much happier to see “228 megabytes” or “\$1.50 per gigabyte”. So in this case you would choose a megabyte as the base unit and a gigabyte as the billing unit.

Supported Service Types

Once installed, PortaBilling® supports the service types shown in the table below. The Rating Base column refers to the applicable rating base options (S means “session-based” and Q means “quantity-based”):

Name	Rating Base	Description
Conferencing	S	Rating conference calls via PortaBridge (or some other conferencing server).
Data Service	Q	Rating data transfers using the amount of data transferred as the billing parameter.
Dialup Internet	S	Dialup Internet access sessions, rated based on session duration.
Internet Access	S, Q	Internet access sessions (DSL, PPPoE, etc.), rated based on session duration or the amount of transferred data.
Messaging Service	Q	Rating messages (text, SMS, MMS, other); charges are based on the number of messages sent.
Quantity-Based	Q	Generic quantity-based service type; can be used to apply charges for any service use expressible in numeric form (e.g. number of pizzas ordered).
Session-Based	S	Generic time-based service type; can be used to apply charges for any service use based on the length of time the service was accessed.
Voice Calls	S	Rating telephony calls (incoming or outgoing) made via PortaSIP®, VoIP gateways or other equipment.
Wi-Fi	S	Wireless Internet access sessions, rated based on session duration.

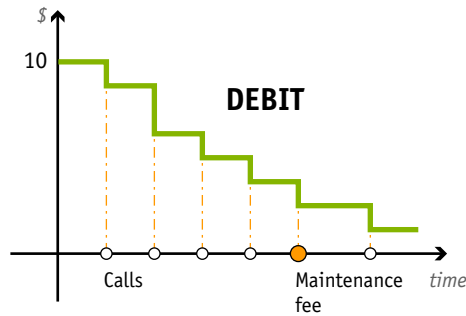
Account and Account Types

An account identifies the end-user who is using the service. For example, a prepaid calling card is an account identified by a PIN, while a customer’s VoIP gateway could be an account identified by its IP address, and a SIP phone is identified by its phone number. There are two main types of accounts (debit and credit) plus an auxiliary one (voucher).

Debit

Typically used for: prepaid calling cards.

- The balance shows the available funds on the account. Initially, the balance is equal to the “opening balance” (typically the prepaid amount), and decreases with every call made.

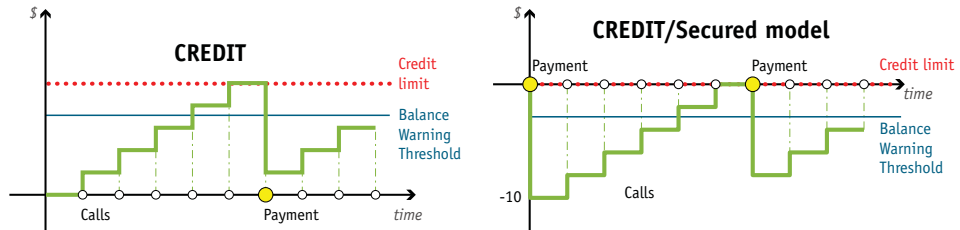


- The account is unusable when the balance hits zero.

Credit

Typically used for: postpaid services.

- The balance reflects how much the account owner owes you. It starts with the “opening balance” and goes up every time a service is used (e.g. a call is made, or a message is sent), and decreases when a payment is received.



- The account is unusable when the balance reaches the “credit limit”.
- The balance can also be negative – this means you owe money to the customer (e.g. the customer has made a deposit). By setting the credit limit to zero you can provide “secured” services (as shown on the diagram above), in which case a customer can only use a service when there is a deposit remaining in his account.
- Unlimited number of simultaneous logins.

Voucher

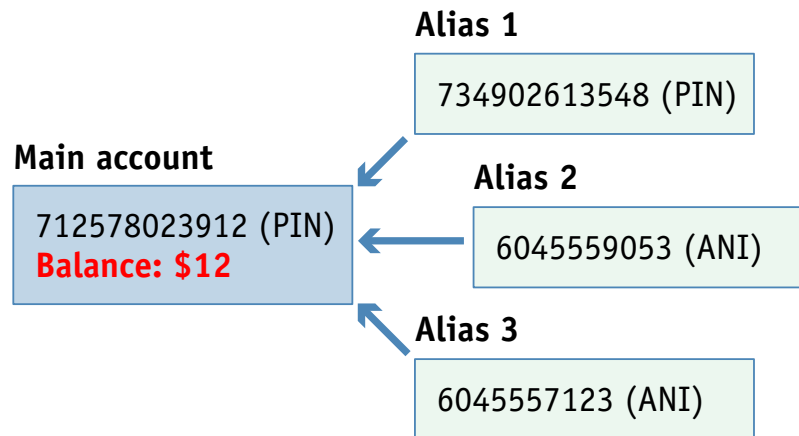
Typically used for: refill debit or credit accounts for a customer using the same account over a long time, e.g. ANI-based authentication.

This can be used to “recharge” (increase the available amount of) a credit or debit account. In this case, the balance of the voucher account is transferred to the main account. An account of this type cannot be used for any services such as making calls. Recharging may be done on the account self-care pages or via a special IVR script.

NOTE: The default Cisco debit card application does not support recharge vouchers, so you must add this functionality to the IVR yourself, or purchase PortaOne’s “Voucher recharge” script by contacting PortaOne sales team. You may also use the voucher recharge IVR provided with PortaUM.

Account Aliases

Sometimes it is desirable to provide multiple means of authentication for the same account; for instance, several phone numbers may be registered for PINless dialing to a single prepaid card. In this case, there will be a single account containing the actual balance and other billing information, and multiple account aliases associated with it. An alias serves as a link or pointer to the main account. When a customer uses an alias ID for authentication, PortaBilling® retrieves information about the main account and uses its balance, product and other parameters for all further operations.



NOTE: When using account aliases in conjunction with debit accounts, PortaBilling® allows only one simultaneous session per main account (regardless of which alias was actually used in the authentication process).

As mentioned above, an alias gives you the ability to use the service via an additional identification, while still drawing on funds from the main account. In some cases you may want to restrict this ability; one typical example is DID forwarding in the case of the SIP trunking service. Here independent registrations for each additional alias created under the main account will be prevented (this is done by deselecting the corresponding checkbox in the alias configuration). An outgoing call can still be made using the alias, if necessary, but this alias cannot register to receive incoming calls independently of the main account.

Account Batches

Accounts can be grouped into batches. Each batch has its own descriptive name, giving you two ways to identify an account:

- By **Account ID** (this is usually some type of ID the customer uses to access the service, e.g. PIN number for calling cards or phone number for residential VoIP services);
- By a combination of **Batch Name** and the account's **Control Number** (a sequential number within the batch).

When creating new accounts, you can either create them in a new batch or add them to an existing batch. In addition to better account tracking, this also permits easy modification of a large number of accounts. You can block or unblock, change the balance, or perform other operations for either the whole batch or a portion of the batch. Note that a batch always belongs to a customer, so you cannot mix different customers' accounts within the same batch, and you cannot have two batches of the same name under two different customers.

You can also create new accounts without assigning them to any batch (this is usually done if you expect to have just a few accounts under a customer). You can always assign a batch to these accounts later.

Account Life Cycle

An account can be created in the inactive state. This allows you to fully provision the whole service configuration for this account, but the service can only be used once the account is activated.

In addition, there are three parameters that define the life cycle of an account:

- Every account has an **activation date**. This defines the earliest date of activation (e.g. you print Christmas promotion cards in November, but set the activation date for December 23rd, so that no one can start using them earlier).
- **Expiration date** defines the date after which the account can no longer be used (e.g. your Christmas promotion cards expire on January 1st). Expiration date is optional, so you may also create accounts which never expire.
- Finally, there is the **lifetime** parameter, which defines the number of days the account remains active after its date of first use. If, for instance, an account has a lifetime of 90 days and is first used on May 1st, then it cannot be used after July 30th. Lifetime is also an optional parameter. If you wish to allow your customers to use the service perpetually, leave this parameter blank.

Lifetime and expiration date work in conjunction – whichever comes earlier defines the end of the service. If an account has an expiration date

of 01-Jan-2007 and a lifetime of 60 days, and is first used on 09-Dec-2006, then the account has only 23 days of service left, and will expire on January 1st.

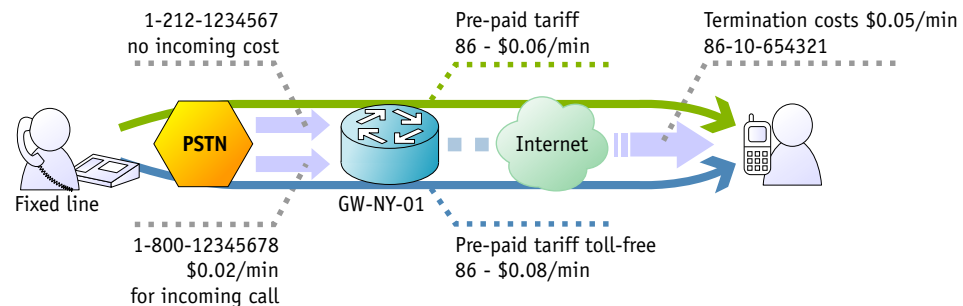
Another important point regarding lifetime is that a voucher recharge operation extends the life cycle of an account. Thus, if an account has a 60-day lifetime, and the account owner makes a voucher recharge 57 days after the date of first use, service will be extended for another 60 days, with the date of the last recharge as the starting date of a new lifetime period.

Products and Accessibility

A product is a combination of services that you provide to a customer for a price. For instance, the same product may provide both dialup Internet access and cheap international phone calls. Or perhaps you will decide to sell calling cards with 6 cents/minute calls to China via a local access number in New York, and 8 cents/minute calls to China via a toll-free line. In this case, your product will include two types of service:

- access via the local New York number, and
- access via the toll-free line,

with price parameters associated with each service.

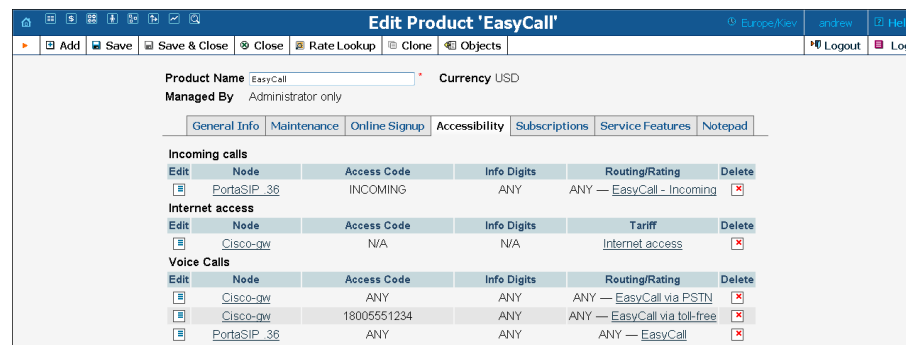


Accessibility is the main component of a product definition. It specifies which services your customers have access to, where on your network they may use these services, and how they should be charged for them. Accessibility allows you to specify the following parameters which define an access point:

1. The type of service provided.
2. The node on which the service is used. What exactly does “node” mean in this context? If, for example, a customer calls to gateway A, enters his PIN, and makes an outgoing call which is terminated on gateway B, is he using a service on node A, node B, or both? The correct answer is that the service is regarded as having been provided **at the point where authorization was performed**. In this example, since PIN authorization is performed on node A, it is node A which must be listed in the accessibility.

3. Identification of the access code (method) on that node. This parameter allows you to use different rate plans for the same service. For example, you may choose a rate plan according to the PSTN access number (local or toll-free) that the customer has dialed. Or you may use different rate plans for outgoing, incoming and forwarded calls in your IP calls service. (While for services such as prepaid cards the access code is a number, for other services any string may be used, so long as it is one provided by the application handling the call).
4. Originating line information (this is applicable only to the voice call service, where the call originates on the PSTN network). You can separate accessibility entries based on originating line information (e.g. whether the call was made from a home phone or a pay phone). Make sure your telecom provider supplies you with this information in the call setup.

Below is an overview of the product bundling and rating possibilities offered by PortaBilling®. The SuperCall product allows customers to make and receive calls from their IP phones, connect to a WiFi hotspot and be charged for time spent online, and make calls as if using a prepaid card (dialing an access number from a normal PSTN phone and then entering an account number and password). You may assign a different rate plan for each service, as shown; for prepaid card services there is even a different rate plan for calls made via a toll-free number or via other (local) numbers.



The screenshot shows the 'Edit Product 'EasyCall'' interface. At the top, it displays 'Product Name: EasyCall' and 'Currency: USD'. Below this, there are several tabs: 'General Info', 'Maintenance', 'Online Signup', 'Accessibility', 'Subscriptions', 'Service Features', and 'Notepad'. The 'Accessibility' tab is currently selected, showing three sections: 'Incoming calls', 'Internet access', and 'Voice Calls'. Each section contains a table of service entries with columns for 'Edit', 'Node', 'Access Code', 'Info Digits', 'Routing/Rating', and 'Delete'.

Incoming calls					
Edit	Node	Access Code	Info Digits	Routing/Rating	Delete
<input type="checkbox"/>	PortaSIP_36	INCOMING	ANY	ANY — EasyCall - Incoming	<input type="checkbox"/>

Internet access					
Edit	Node	Access Code	Info Digits	Tariff	Delete
<input type="checkbox"/>	Cisco-0w	N/A	N/A	Internet access	<input type="checkbox"/>

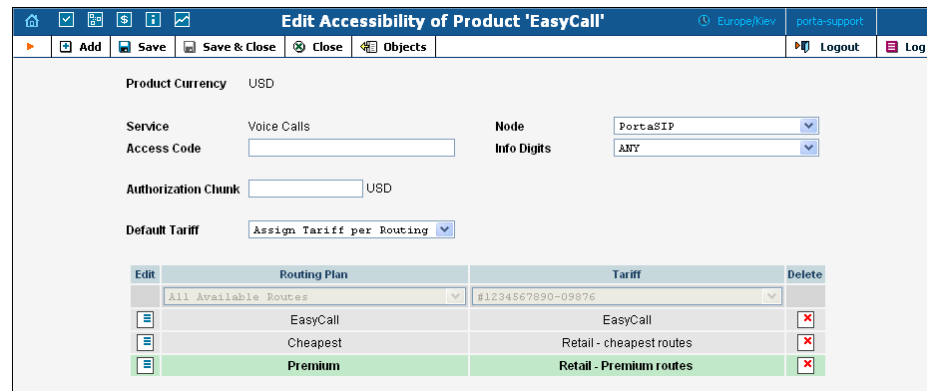
Voice Calls					
Edit	Node	Access Code	Info Digits	Routing/Rating	Delete
<input type="checkbox"/>	Cisco-0w	ANY	ANY	ANY — EasyCall via PSTN	<input type="checkbox"/>
<input type="checkbox"/>	Cisco-0w	18005551234	ANY	ANY — EasyCall via toll-free	<input type="checkbox"/>
<input type="checkbox"/>	PortaSIP_36	ANY	ANY	ANY — EasyCall	<input type="checkbox"/>

Applying tariffs based on routing plans

Normally, the customer is charged the price which is pre-set in his price list. Naturally, this price is calculated based on estimated average termination costs, but does not directly depend on which carrier is used for a specific call. Since you may have different termination vendors with various prices and levels of audio quality, how can you give your customers the freedom to choose whether to make phone calls via low-cost carriers (and thus pay low rates), or use high-quality carriers (and so pay premium rates)?

Routing plan selection is available for services based on the **Voice Calls** service type. It allows you to select a tariff for charging a customer based

not only on the “incoming” point to your network, but also on the “outgoing” entry, i.e. the type of route used to terminate a call. Each type of routing (low-cost, premium, etc.) is defined as a separate routing plan, and multiple routing plans can now be associated with a single accessibility entry in the product definition. A specific tariff will be assigned to each routing plan, and this tariff will be used when a customer subscribing to a specific plan places a call. For more details, see the *Routing Plan Selection* chapter.



You can still use the “single tariff per accessibility entry” mode. Since this provides faster performance, it is a good idea to use it when the real-time selection of routing plans is not required. In this case, you can still assign a default routing plan to an account, and this will be used when the end-user makes a phone call.

“Single tariff” mode is active by default when you create a new accessibility entry. If you wish to apply a specific tariff based on the routing plan used, you have to manually select the “Assign Tariff per Routing Plan” option in **Tariff** select menu.

Maintenance Fees

Maintenance is a property of the product which allows you to charge your customers periodic fees. A maintenance fee is applied to all active accounts (accounts which have been used at least once) for a specific product. A maintenance fee is applied for the first time on the day after an account is used for the first time, and at periodic intervals thereafter. If you change the maintenance fee for a product, this will affect maintenance fees that are charged in the future (but not past ones). It is possible to temporarily disable maintenance fees and then start charging them again from a certain date.

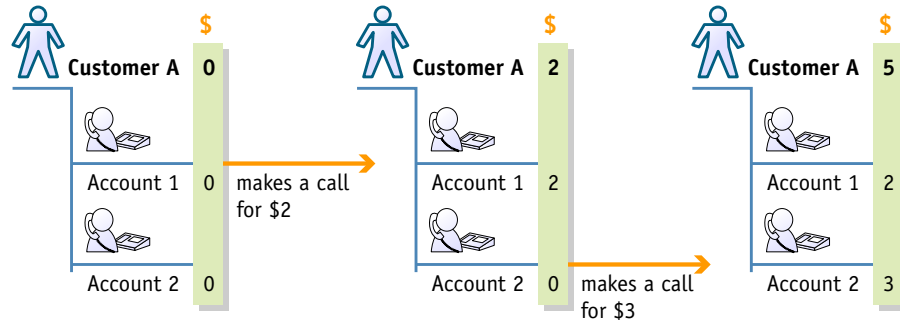
NOTE: Maintenance fees remain an important tool in the prepaid card business model. For subscription fees in a postpaid scenario, a new method is available starting with Maintenance Release 12: subscriptions. Please see below for more details.

Customers

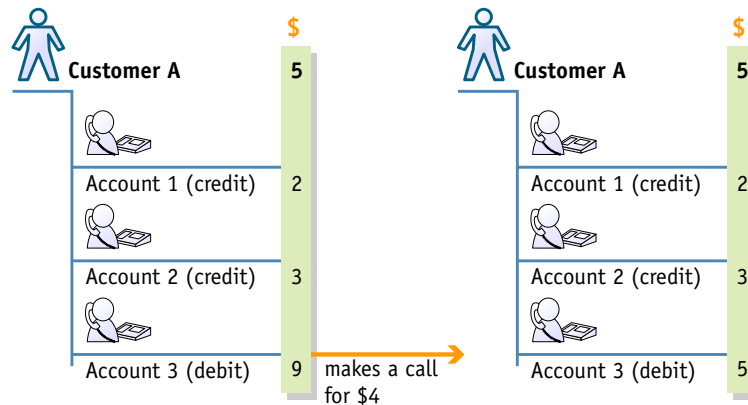
Retail Customers

A customer is an entity (an individual or a company) who is responsible for using the service via one or more accounts. For instance, when a family signs up for residential VoIP services, a man, his wife and his brother will all have individual phone lines, each represented by an account. The person who signs the contract (and who is responsible for paying the bills) will be entered as the customer, and all the accounts will be listed under him.

When a credit account is charged for using the service, this directly affects the customer's balance, and the activities of all accounts are included on the consolidated invoice. This allows quick tracking of service use for a group of accounts, or the application of a credit limit for the whole group.



NOTE: If a call was made by a debit (prepaid) account, this does not affect the retail customer's balance (see below), since prepaid accounts (cards) are normally paid for by your customer at the time of purchase. This also allows your prepaid card distributor to have postpaid accounts (for example, for internal office use) and to be billed for them. Or, your postpaid customer with multiple IP phones can purchase additional prepaid cards for his temporary workers.



Typically used for: business end-customers (when more than one account per customer is required)

Distributors

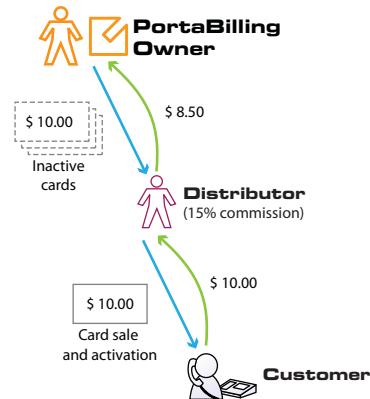
The distributor model is designed to expand sales activities by engaging additional agents and enlarging the point-of-sale network, without any significant costs or risks.

Customers can purchase new products, settle their invoices or refill their prepaid accounts by paying cash to a distributor (for example, the owner of the grocery shop opposite your house could be a distributor). After the distributor collects the money, he delivers it to the ITSP, minus his commission. PortaBilling® automatically calculates the commission and checks the distributor's balance to keep track of how much he owes the ITSP, and also to verify the distributor's credit limit. This helps to avoid a situation where the distributor would activate too many accounts without first submitting payment for accounts already activated, and thus limits the ITSP's risk of loss in case the distributor goes out of business. A distributor need not have any special technical knowledge or skills. He either delivers to the end-user a tangible product (e.g. calling card) supplied to him beforehand in bulk, or, in the case of an intangible product (e.g. registering a customer's cell phone number so he can make PINless dial calls), he can create accounts/customers using a quick form. All account management operations are done via the web interface.

While a distributor might only sell a few new accounts each day, creating accounts in such small amounts at the distributor's request requires too much administrative overhead. Also, when prepaid cards or top-up vouchers are being printed in the print shop, it makes sense to do this for many thousands of cards at once.

Therefore, the account generator can be used to produce a large batch of accounts so that all the cards can be printed at once. Then later a distributor can be assigned to some of these accounts (i.e. when this distributor receives a portion of cards to sell).

Nonetheless, when some of a batch of cards is delivered to a distributor their total face value could be quite large. So it is risky to allow all of the cards to be active (e.g. if the cards are stolen, they could all be used). Thus cards (accounts) are typically generated and supplied to the distributor in an **inactive** state, and only when the distributor activates the card (account) can it be used to access the service. Likewise, the distributor is only charged when activation is done.



The following are associated with a distributor:

- **Default Sales Commission** (applied when an account is activated or a distributor is assigned to an active account);
- **Default Payment Commission** (applied when a payment is entered by a distributor).

When a customer or account is created or activated under a distributor, the latter's balance increases by the account's balance, minus his commission. For example, if the distributor's commission percentage is 15% (the default sales commission) and he sells a new account with a balance of \$10, the distributor is charged \$8.50 (this is the amount he owes to the ITSP). So the distributor receives \$10 from the customer and gives back \$8.50 to the ITSP, thus making a profit of \$1.50. If the distributor applies payment of \$10 towards an account, and his payment commission percentage is 10% (the default payment commission), he is charged \$9 and receives a \$1 profit.

Thus the distributor is charged when:

- he applies payment towards a customer or an account;
- he is assigned to an active account;
- an account to which he is assigned has been activated, or a new account is created in the active state.

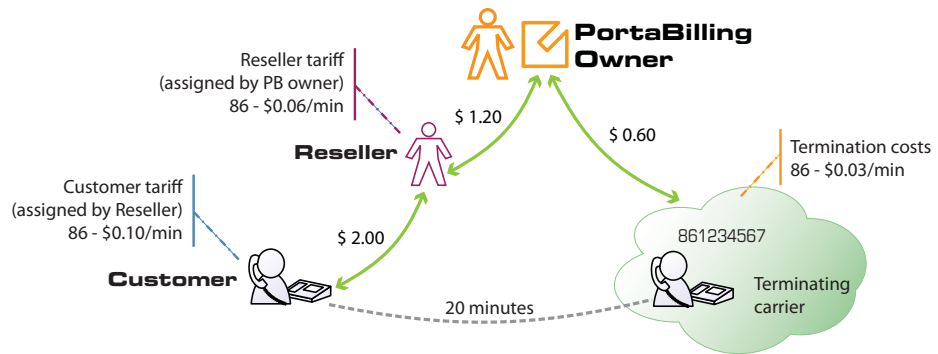
To prevent operator error when attempting to generate accounts being assigned to a specific distributor, the account generator will only allow the creation of inactive debit accounts/vouchers, and these accounts will have to be activated later.

Typically used for: Point-of-sale agents, where an agent delivers the product to end-users, collects payments, and then delivers the money (minus his commission) to the service provider.

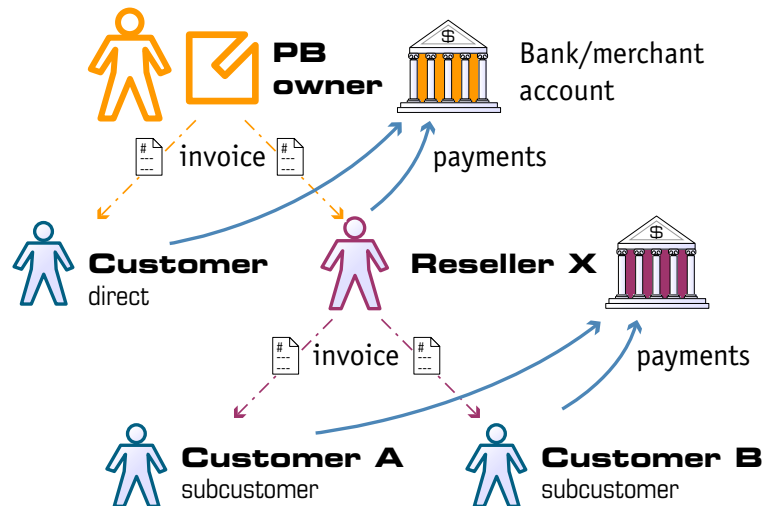
Resellers

A reseller is a partner who resells your services, and for whom you perform service provisioning and the billing of his subscribers. He owns retail customers (also called **sub-customers**), who in turn own individual accounts.

When a call is made by an account (end user), the reseller is billed using his own tariff (wholesale rates). Thus the end user (account+retail customer to whom it belongs) and reseller are billed separately; they may even be in different currencies. So if, for instance, a partner is reselling your services, he will pay you 0.06 USD/min. for each call made to the US by his subscribers. At the same time, he can charge his customers any rate he wishes (e.g. 0.10 USD/min).



Reseller acts as an independent service provider: he will be able to assign his own rates to his subscribers, PortaBilling® will automatically generate invoices for his customers with reseller's name on them, sub-customers may go to the specific domain to access their self-care interface and see reseller's name and logo on these pages, etc. Reseller collects money from his sub-customers and is responsible for paying invoices, issued by the PortaBilling® owner. If reseller exceeds his credit limit – this automatically blocks all activities of his sub-customers.



Typically used for: OEM or “white-label” service providers, companies who would like to offer the service under their own brand name, while the actual service is provided on your VoIP network and all the billing and management of their customers is done in PortaBilling®.

NOTE: Billing via the reseller model implies an additional load on the billing engine, compared to a retail customer, and using only retail customers is about 50% faster than when a reseller is involved. If performance is an issue, avoid creating resellers when it is not necessary.

Summary of customer types

There are the following types of customers (all of whom have the same attributes, such as balance or address info):

- Reseller – sells the service provided on your platform under his own name and product brand;
- Distributor – resells your products to end-users;
- Retail customer – someone who actually uses the service.

In addition, all customers can be subdivided into two categories:






- Direct customers – Customers who directly communicate with your company, i.e. receive bills from you and pay to your accounts. A direct customer can be a reseller, a distributor or a retail customer.
- Sub-customer – A retail customer or a distributor who does business with a reseller (and in PortaBilling® terms, belongs under that reseller).

Retail customers, distributors and resellers are billed according to the credit model. Their balance reflects the amount they owe you. Thus it starts from a certain value (typically 0) and goes up with each call made, or down with each refund or payment. If the customer’s balance reaches the

maximum credit amount (credit limit), his accounts will not be allowed to use the service further. The only exception is a retail customer’s debit accounts: since these are prepaid, and their calls do not affect the customer’s balance, they can still make calls even if the customer has exceeded his credit limit.

Customer status

A customer may have the following status:

Status	Description
 Credit exceeded	This indicates that the customer’s balance is above his credit limit, so he will not be able to make any outgoing calls, unless these are free calls (e.g. toll-free numbers).
 Payment frozen	Similar to the item above, this indicates that the customer’s periodical payment has been suspended due to repeated errors (e.g. credit card cancellation).
 Blocked	The customer has been blocked by the administrator; no call services will be provided until the administrator removes the block. The customer still has access to his self-care pages. Blocked customers are not subject to maintenance charges, but subscription fees still apply.
 Suspended	Services for this customer have been suspended because of an overdue invoice. Once payment is received (whether the customer makes an online payment, or a periodical payment is applied, or the administrator enters a cash/check payment manually) in an amount covering the overdue invoice, the suspension is automatically lifted.
 Closed	The customer’s account has been closed, and is kept in the database only for informational/regulatory purposes. No further operations are possible with this entity.

Customer Classes

Customer classes allow you to define a set of parameters which will be shared among a certain category of customers. For instance, suppose your invoice term for retail customers is “Net 21 days”, while for business customers it is “Net 30 days”. If your operators enter these values manually for each customer, there will inevitably be mistakes. Instead, you can define two separate customer classes, one named “Retail” and the other “Business”, and define these parameters within them. After that, your operators need only assign a specific class to a given customer in order for the customer to automatically inherit all the class properties (grace period, invoice template, and so on).

Vendors

Vendors are your service providers, for example termination partners or incoming toll-free line providers. Every time a call travels from your network to a vendor (via telephony or VoIP), a cost is associated with it. So at that point PortaBilling® will charge the account and customer for the call, and also calculate your termination costs.



Currently, vendors are active **only** in the “Voice call” service, since for other services (e.g. WiFi access or dialup) there are no direct costs associated with a particular session that are to be recorded as an individual xDR (for instance, your uplink provider might charge you a flat monthly fee for a 1Mbit/s channel to your WiFi hotspot).

Connections to a Vendor

Connections define points where calls travel from your network to a vendor. (Unless it is a direct call between two users on your network, there is always a connection to a vendor.) For connections via IP, we need to know the IP of the remote node (where the call is being sent to), while for telephony we need to know the gateway and, optionally, the name of the port the phone line is connected to. A connection defines a termination cost, i.e. the tariff according to which a termination partner charges you.

Connections from a Vendor

It may happen in your business that you incur costs not only when terminating calls, but also when a carrier delivers calls designated for one of your customers to your network. Typical examples of situations in which such charges occur include the following:

1. You purchase a toll-free number from a carrier and interconnect with it via E1/T1. When a customer dials this number, the call is delivered to your gateway, and you can then provide a service (e.g. a prepaid card IVR) to the customer.
2. You purchase a set of phone numbers (DIDs) to be distributed to your customer's IP phones. When a number is dialed anywhere in the PSTN world, it will travel to the carrier who owns these numbers. The carrier will then forward the call either directly to your SIP server via an IP, or to your VoIP gateway via a PSTN trunk.

In both cases, you will be charged by the carrier who provides you with the service according to call duration, and these expenses must be reflected in PortaBilling® in order to show your cost/revenue figures correctly.

NOTE: Account and customer are billed **only** for outgoing connections. Connections from a vendor are purely for calculating costs.

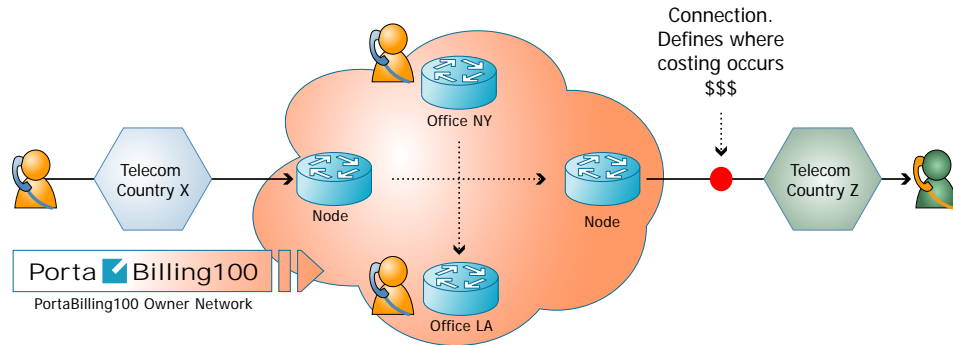


Figure 2-1

Connection Load

PortaBilling® can build load graphs for every connection, so that you can monitor average load and quality parameters (e.g. ASR) for each one.

Account ID-based Billing

Modern business involves diverse types of services, which require different types of billing. Calls made by prepaid cards are billed by PIN number, calls from a customer’s GW should be billed by IP address, some wholesale customers are identified only by the port they are connected to, and so on.

It is easy to find out that a billing event occurred (e.g. a call was made). The important thing is to correctly determine who made the call and, therefore, who should be charged for it. PortaBilling® uses a simple and flexible yet powerful method to determine a call’s owner. Identification of the account (Account ID) is sent to the billing system in the **User-Name** attribute.

For example, in the case of prepaid calling cards, **User-Name** will contain the PIN, IP address for IP-based billing, and so on. Of course, an account with such an account ID must exist in the database. This way of utilizing the **User-Name** attribute is actually default behavior for Cisco call handling applications, and the latest firmware releases of Quintum should do the same.

Since PortaBilling® supports account aliases, in some cases the value of the **User-Name** attribute will first be matched against the ID of an alias, which then will be used to retrieve the main account information. However, the principle remains the same.



Due to such an approach, logic in the billing is independent from service-specific issues. You are not limited to any pre-defined billing schemes in PortaBilling®; rather, you can design your own. Today you can bill by PIN or IP address, and tomorrow you will additionally be able to bill by h323-id. All you need to do is pick the proper application to handle the call, so that a correct identification will be sent in the `User-Name`.

Unfortunately, for some types of service `User-Name` is not sent to the billing properly on Quintum gateways. In this case, it is possible to employ username replacement rules on the billing side to emulate this. See the relevant section in the "How to:" chapter for more details.

Nodes

Nodes are your gateways for facilitating calls, providing Internet access via WiFi, sending messages, or supplying any other type of service to the end user. Perhaps the most important feature of a node is that it can ask the billing for authentication or authorization and send accounting information to the billing system. It is very important that we exchange AAA data **only** with trusted gateways, also called **trusted nodes**. Usually, if this gateway is not owned by your company, but rather by your partner or customer, it will not be considered trusted, and so will not be created as a node in PortaBilling®.

Node ID, NAS IP Address, and Radius Source IP

What is the difference between these terms?

- **Radius Source IP** address is the address that radius requests (UDP packets) come from. Radius will accept requests only from those IP addresses which are listed as **Radius source IP** for one of the nodes. This is used as protection against denial-of-service attacks and attempts to send fraudulent information to the billing system.
- **NAS-IP-Address** is the IP address gateway used for VoIP purposes. This address will be present in the radius data as the **NAS-IP-Address** attribute, and is used to identify which node sent the request. Usually **Radius Source IP** is the same as **NAS-IP-Address**, except in a situation when your gateway has two network interfaces, using one (internal) to communicate with the billing system and the other (external) for VoIP traffic.
- All of the incoming VoIP calls should be authenticated (see **Remote IP authentication** below). This also applies when a call comes from your gateway A to your gateway B. When node B (the

terminating node) consults PortaBilling® as to whether the call should be allowed, it will send an authentication request to PortaBilling® with a User-Name containing some identification of the remote gateway. What information is used for identification depends on the application, but typically this is the IP address of the remote gateway. PortaBilling® will check if there is a node with such a **Node ID**, and, if so, the call will be allowed.

NOTE: You can only specify your own custom **Node ID** if you have enabled this advanced feature. Normally this is not required, and by default **Node ID** equals the **NAS-IP-Address**.

In most cases, all three parameters (NAS-IP-Address, Radius Source IP and Node IP) are the same.

Virtual Environments

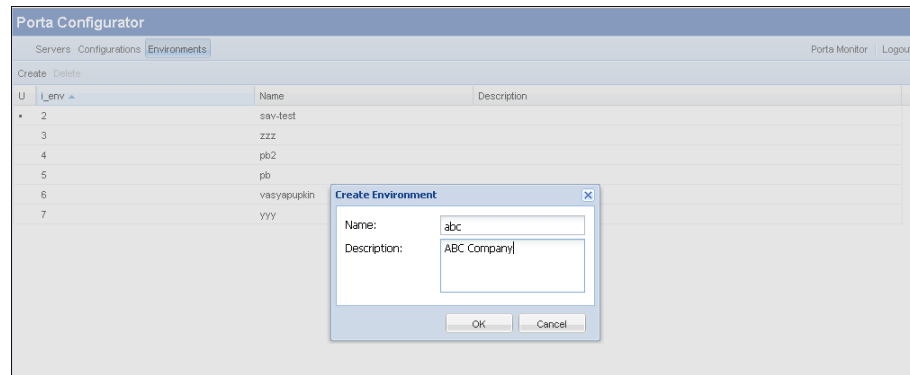
It is possible to have multiple isolated PortaBilling® instances, or virtual environments, within the same PortaBilling® installation. This functionality allows you to use PortaBilling® as an ASP (Application Service Provider) platform, or to create test environments which do not affect billing production activities.

Each PortaBilling® virtual environment contains a complete set of data specific to that environment. A user who logs in to a user account for a specific environment will see only the users, customers, tariffs, accounts, xDRs, statistics, and so on for that environment. There are a few restrictions on the use of virtual environments to ensure their correct functioning:

- The login name for Admin users must be unique across all environments. For example, you may not have a user 'jack' in both environment ABC-Production and environment ABC-Test. Rather, two users will have to be created, possibly 'abc-jack' and 'test-jack'. The same applies to login names used by your customers/accounts/vendors on their self-care pages.
- A gateway (node) may be registered to only one environment. So you cannot have a node with the same IP address in two environments.

You may have multiple customer, vendor or account entries with the same name across environments. For example, you may have an account '12345' in both 'ABC-Production' and 'ABC-Test' (from the previous example). In this case, the decision as to which account (from which environment) is determined by which gateway sends the accounting record (i.e. one registered in ABC-Production or in ABC-Test environment).

You may create new environments or delete them from the configuration server web interface.



A new environment is created empty, except for a single user account which you will use to login and then create other users, tariffs, customers, and so on. By default, the username of this user is `<env>-root` and the password is the same, where `<env>` is the name of the environment you have just created. So if you create the environment **abc**, to log in to it you will use the username `abc-root` and the password `abc-root`.

3. Rating and Invoicing

Destinations, Rates and Tariffs

Destinations, rates and tariffs are the essential parameters which define how a certain event should be billed. It is very important to find a correct set of billing parameters and rules (rates) for every event, and this is done by matching a service identifier with the available rates in the rate plan (tariff). The service identifier specifies exactly how the service was used. For example, the service identifier for voice calls would be the destination number (Called-Station-Id, or CLD, or DNIS). For SMS messages, it would likewise be a destination phone number, while for services such as a pay-per-view movie it could be the movie category (“New Release”, “International” or “Classic”). Since the rating of telephony numbers is the most complex case, it will be the primary focus of this chapter.

Destinations

Destinations are a list of all possible phone number prefixes to be used in your system. You generally need a new phone prefix when you have a new service area for calls which are to be treated differently than others. For example, if you start providing calls to the Czech Republic, you should add destination 420 and specify it as Czech Republic. Later, if you plan to charge calls to Prague differently than calls to the rest of the Czech Republic, you might need to add another destination with the phone number prefix 4202. All such destinations have to be entered into the system before you can use these prefixes. This prevents errors and helps you to improve data quality. It is recommended that all of your prefixes be defined in the E.164 format.

It is virtually impossible to have an “ultimate” destination list that would contain all prefixes for the entire world. First of all, it is quite difficult to gather and maintain such information. Second, and even more important, is the fact that having “all possible” prefixes would not offer us any real benefits, and would only make rate maintenance more difficult. For instance, if vendor A provides you with a rate of 0.10/min for calls to 420 (Czech Republic) and a 0.18 rate for calls to 420602 and 420603, you will need to create three prefixes: 420, 420602 and 420603. If you also create the prefix 4205 (Czech Republic, Brno) it will only take up space in the database, as it will not ever actually be used.

Therefore, we recommend that you follow a “required minimum” approach: only create those destinations for which you will need specific rates for your carriers or customers.

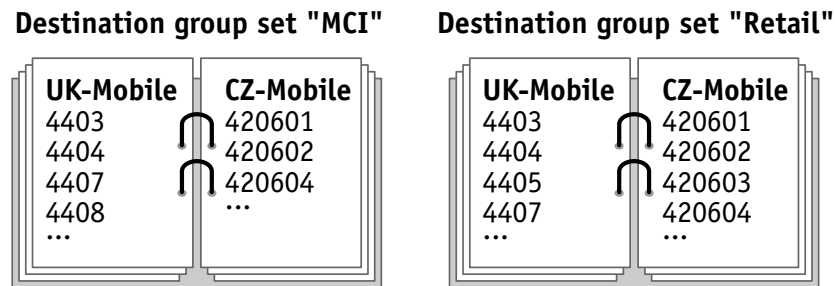
For service types other than those based on phone numbers, you may create symbolic destinations such as WIFI, NETACCESS or MOVIE-BLOCKBUSTER.

Destination Groups

Sometimes you will have several prefixes for the same target destination, e.g. 420602, 420603 and 420737 for mobile numbers in the Czech Republic. All of these prefixes can be defined as the single destination group “CZ Mobile”, so when you enter a new rate for this destination group it will, in effect, create rates for each individual prefix.

Destination Group Sets

It happens quite often that different partners will assign a different meaning to the same destination groups. For instance, your partner A says that “CZ Mobile” is 420602, 420603 and 420737, but your partner B also considers prefix 420777 to be part of “CZ Mobile”. So you will need to have two versions of “CZ Mobile” – one for partner A and one for partner B. In this situation, you will create two separate destination group sets (“A” and “B”), as well as different destination groups inside those sets.



You can imagine each destination group set as a binder, with every page in that binder describing a certain destination group.

Rates

A rate is a combination of billing parameters for a specific destination. For example, you can specify that calls to 420 are charged 0.09 USD/min during peak time and 0.07 USD/min during off-peak time, while calls to 420609 should not be allowed at all.

To easily manage the process of routine rate changes, each rate in PortaBilling® is assigned an exact time when it comes into effect (the **Effective From** attribute). This allows making an automated rate change in the future, so that a new rate can be entered today, but used only after midnight on January 1st. When an existing rate is edited, this does not

replace the actual rate, but simply creates a new rate effective from this moment, so that the PortaBilling® administrator can always browse the history of rate modifications.

Tariffs

A tariff is a complete set of rates for a specific account, customer or vendor. Thus it should include every possible destination allowed by their service plan (e.g. in the case of a telephony service, every destination to which you want to let them call).

It may be that the tariff will contain some prefixes that are part of more generic ones (for example, you will have a rate for the **420** prefix and for the **4202** prefix). In this case, the longest prefix match takes priority.

What happens if a destination is not included in the tariff, and the customer tries to call there? There are two possibilities:

- If outgoing calls are authorized via PortaBilling® (for example, by PortaSIP or by the prepaid card's IVR) the customer will not be authorized to call this destination.
- If the gateway or switch which processes the call does not perform authorization in PortaBilling® before allowing the call to pass through, the customer will actually be able to make the call, but PortaBilling® will not be able to bill it properly, since the required information is missing. An email alert will be sent and a special CDR will be written into the database, so you will still have an overview of this call. It is highly recommended that you always use authorization of calls via PortaBilling®.

Wildcard destination

There is one special destination: | (pipe). When a rate for this destination is present in the tariff, it will match any dialed phone number. But this will only happen, of course, if there is no other, better match by the actual phone prefix. Such a rate should probably not be used in the customer's tariff, since it effectively authorizes him to call any phone number in the world. It is very useful, however, in tariffs for internal purposes. For instance, we can use it in the tariff used to produce CDRs for SIP-to-SIP on-net calls, since there the cost is zero regardless of which number is actually dialed. Otherwise, that tariff would have to include all phone prefixes for all possible phone numbers assigned to IP phones.

Also, for some services or billing features you may need to create other destinations. For example, in order to use the "Voice VPN Rating" features, which charge calls between extensions in the same IP Centrex environment according to a pre-defined rate, you will need the destination VOICEVPN. Consult the reference manual for a particular feature to find out which additional destinations it may require.

Relationship between destinations, rates and tariffs

Let's move from VoIP to a simpler scenario. Imagine you are the owner of an office supply store. You have to manage your inventory and price lists for customers and resellers.

- First of all, you must create a catalog of all the items you intend to sell. This is your internal document containing entries such as “20001 – ball-point pen, blue; 20002 – ball-point pen; black; 50345 – stapler;” and so on. Note that this is just a description of the items, without prices, since the price will depend on the specific vendor/customer.
- You will receive price lists from your suppliers and convert them into data files for every vendor, such as “Office Depot: 20001 – \$0.35; 20002 – \$0.35; etc.”. Note two things: there is no need to include an item description in every file – you can always extract this from the catalog of items you have created above. Also, each data file will contain only some of the items, i.e. those which are provided by this particular supplier.
- The next step is to create similar price lists, but with the prices you will apply to your customers. Of course, you might have several such price lists: e.g. one for retail customers, another for business customers, another for resellers, and so on. Every data file will contain all of the items you offer to this category of customers, including prices.

Now let's come back to the VoIP business:

- *The catalog of items* corresponds to **destinations** in PortaBilling®. Every destination is similar to an item description in the catalog, i.e. it provides general information (e.g. 420 – Czech Republic proper, 420602 – Czech Republic mobile, and so on).
- *The price list (for vendor or customer)* is equal to a **tariff** in PortaBilling®. The price list includes all items the customer may purchase and all destinations the customer can make calls to. All tariffs are identical in structure, but some of them will be used to calculate your costs (vendor tariffs), while others will be used to charge your customers.
- *A single line item in the price list* is equivalent to a **rate** in PortaBilling®. It gives the price per minute and other rating parameters for a specific destination.

Therefore, the standard sequence for setting up your service is as follows:

- Define the destinations you will use in your business. Basically you will need to define every unique prefix used by your vendors (an easy way to do this is by using PortaBilling®'s default destination set and PortaBilling® templates).
- Create a tariff for every vendor using the rate list they supply.

- Create a tariff for every customer/product. Once again, note that each tariff may contain a different set of destinations. For instance, the tariff for your vendor ABC may contain different rates for 420 (0.10/min), 4202 (0.09/min) and 420602 (0.18/min). But you can just list a single rate for your customer in the tariff, i.e. 420 – 0.20/min.

Relationship between tariffs and destination group sets

As mentioned before, multiple destination group sets may be defined in the system, with some destination groups (e.g. CZ-Mobile) defined for each of them. So, when you enter a new rate for CZ-Mobile in a specific tariff, which destination group definition should be used? To avoid such ambiguities, you will assign a destination group set for every tariff, which will be used to create new rates (if you do not assign a destination group set to the tariff, then you can only enter rates for individual prefixes). Thus when you attempt to create a new rate for destination group CZ-Mobile, the following sequence of events takes place:

- PortaBilling® locates the definition of the CZ-Mobile destination group in the destination group set associated with this tariff.
- For every destination included in this destination group, the new rate is inserted.

In other words, the result is the same as if you had created multiple rate entries manually. Rates which are created using a destination group do not differ in any way from rates created for a single destination. One important consequence of the foregoing is that if you change the destinations included in a destination group, it will not affect the previously created rates. Thus if you had 420602 and 420603 in the CZ-Mobile destination group, and you now add 420737, this will not affect any of the tariffs. In order to have correct billing for the 420737 prefix, you must go to the corresponding tariff and add a new rate for the CZ-Mobile prefix.

Billing for Always-on Services

Always-on services are those which are consumed without interruption over extended periods of time. The only thing that varies is the rate of consumption. For instance, a computer connected to the Internet is constantly transferring data. However, the amount tends to be very low at night, when no one is using the computer, and higher during the day, when emailing or web browsing are taking place. It can even be extremely high when content like video-on-demand is being downloaded.

The biggest challenge when charging for services such as data downloads is that, although the customer uses the service all the time, some specific, “finite” entries must appear on his invoice, e.g. “Data downloaded May 1st

- May 31st”. Also, customers expect to see real-time information about the charges applied, as in the case of other types of services.

Normally, PortaBilling receives information about service usage via “keep-alive” (or “update”) requests. Each of these informs billing that the session is still in progress and what the current service consumption is (e.g. the amount of currently downloaded and uploaded data). For each such request, PortaBilling determines which billing session it relates to. For example, although from the customer’s point of view the session did not terminate at midnight on June 1st, this was nonetheless the end of his billing period, and a new billing session has started, with charges accumulating to a new xDR. When processing an update request, PortaBilling calculates the new amount to be charged for session activity up to the present moment, and updates the xDR for the session accordingly. This billing method, i.e. constantly modifying an xDR record to reflect the new charged amount, is quite different from the one used for other types of services, where PortaBilling receives the information only after the service has been completely consumed, and so the xDR is created only once.

PortaBilling’s open architecture allows to process virtually any type of “always-on” service. Take electricity billing, for example: your household consumes electricity non-stop, but consumption is lower during the night (when only the refrigerator is running), higher during the evening (when the lights are on), and extremely high during the day (when various appliances are in use).

Session Billing Parameters

Time-based services such as WiFi access, where the cost depends on the total session duration, are rated exactly the same as voice calls, as explained in the following section.

Call Billing Parameters

Cost-based and volume-based billing

The traditional method of billing is cost-based billing. This means that, when a call is completed, the billing engine calculates the charges for the call based on price parameters for the call destination and duration. This amount is then applied to the account, customer or vendor balance. Volume discount plans allow rating of the call based not just on information about it, but can dynamically adjust billing with regard to other calls already made (e.g. if you have made more than 500 minutes of calls to US&Canada this month already, the current call will be charged at a 10% discount to the standard rate).

Cost-based Billing

Peak and off-peak prices

It is possible to have different sets of prices for peak and off-peak time. Off-peak periods are defined using the powerful and flexible Time:Period module. An Off-peak Period Wizard is also available, allowing you to perform period definition the easy way. See the [PortaBilling® 100 Web Reference Guide](#) manual for more details.

The fundamental concept here is the period definition, which specifies a certain interval in time (typically a repeating one). In each period you can specify conditions for:

- Time of day
- Day of week
- Day of month
- Month

If you do not specify a condition for a certain element (e.g. for **Month**) then it is considered that there are no limitations on this element, e.g. “from 8pm to 8am” will apply in the same fashion throughout the year. If you define conditions for several elements, these must be satisfied at the same time, e.g. “from 8pm to 8am, Monday-Friday” means that an event fits into this interval if it happens between 8pm and 8am **and** from Monday through Friday, e.g. 6am Saturday is excluded.

An off-peak period can be a combination of several definitions, so that a given moment is considered to be within the off-peak period if it satisfies *at least one* of these definitions. For instance, for the period “from 8pm to 8am, Monday-Friday OR Saturday-Sunday” both 9am Saturday (since it satisfies the second condition) and 6am Friday (since it satisfies the first condition) are included.

By default, everything is considered to be peak time. You may use one of the three available options to determine whether a peak or off-peak rate should be used. When defining an off-peak period, you may choose for off-peak rates to be applied if:

- a call starts during off-peak time
- a call finishes during off-peak time
- a call both starts and finishes during off-peak time (the least flexible option)

Multiple off-peak periods

It is also possible to offer two distinctive off-peak periods (e.g. different prices for calls made overnight or on weekends). You can define two independent off-peak periods by enabling the `Multiple OffPeak Periods` configuration option. In order to add a second off-peak period definition

in the Off-peak Period Wizard, create your first off-peak period, then use the **Add another period** button. In order to use different rates for two off-peak periods, you must use a call rating formula. When defining the formula, select the special **pricer** element in the rating formula as **price per minute** instead of **price_first/price_next**. PortaBilling® will then check whether the call fits into the main period; if so, **pricer** will be set to the **Off-peak price next** value. If the call does not fit into the main period, then the alternative period will be checked; if the call fits into this period, **pricer** will be assigned the **Off-peak price first** value. Otherwise, peak rates are applied, and **pricer** will be populated with the value from **Price_next**.

So, for instance, in order to use the 0.10 rate during peak hours, 0.08 during weekends and 0.06 during night time, you should specify the main off-peak period as “Saturday and Sunday” and the alternative one as “8pm-8am”, construct a formula such as “N intervals of 60 seconds at pricer USD/min”, and enter the following in the rate data:

- Peak price_first, price_next = 0.10
- Off-peak price_first = 0.06
- Off-peak price_next = 0.08

(Yes, it is correct – the **main** off-peak period price is specified in the **price next** parameter).

Charging calls – traditional method

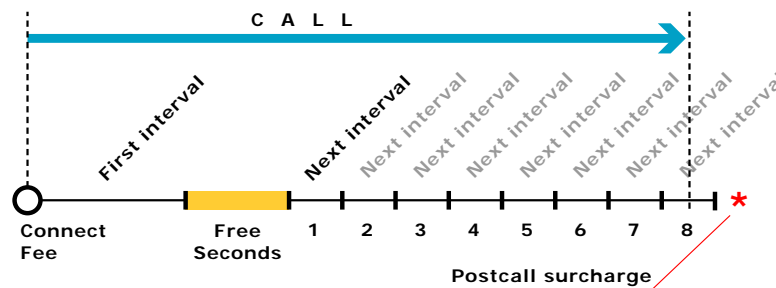


Figure 3-1

The figure above demonstrates how calls are charged. A **Connect Fee** is charged immediately upon connection, and all calls shorter than the *First interval* will be rounded to **First interval** seconds. Free seconds are granted after the *First interval*, so this part of the call is not charged. Calls longer than (*First interval*+*Free seconds*) will be rounded up to multiple **Next interval** seconds. After that, the **Post Call Surcharge** is applied. The call illustrated in the figure above will be charged using the following formula:

$$\text{Amount_Charged} = (\text{Connect_Fee} + \text{First_Interval} * \text{Price_First}/60 + 8 * \text{Next_Interval} * \text{Price_N}/60) * (1 + \text{Post_Call_Surcharge}/100)$$

Parameters such as **First interval**, **Next interval**, **Price First** and **Price Next** can be specified per destination. **Connect Fee**,

Free Seconds and **Post Call Surcharge** are defined on a per-tariff basis, and so will be the same for all destinations within a tariff.



Attention: The billing unit may be any length, but price must always be entered on a per-minute basis. This allows better operations with tariffs, for example, comparing two tariffs with different billing intervals or entering rates into the system from an external source.

Charging calls – rating formula method

This method gives you maximum flexibility in call rating. You create a call formula which is then applied to the call. This formula can consist of an unlimited number of charge elements (the only limit is the length of the formula text). These elements are:

- **Interval** - charged part of remaining call duration; the parameters for this element are:
 - **Duration** – rounding period (in seconds)
 - **Count** – number of rounding periods in the interval
 - **Price per minute** – automatically prorated according to the rounding period duration

The charge for the interval is calculated as:

$$\text{Total} = N * (\text{Duration}/60) * \text{Price_per_minute},$$

Where N is either equal to Count (if Call_Duration is more than Count*Duration) or else Call_Duration/Duration (rounded up).

Thus you can create intervals such as “First 10 minutes – price per minute is \$0.05 and we will round up to 30 seconds” – in this case, the duration will be 30, the price per minute 0.05 and the count 20 (because there are twenty 30-second intervals in 10 minutes).

- **Fixed Surcharge** – fixed amount to be added to the total call cost
- **Relative Surcharge** – the total call cost (at this moment) will be increased by X%

Formula elements are applied until there is no more non-charged call duration left. The surcharges which follow a certain interval are applied only if this interval has been fulfilled. “Fulfilled” means that the interval is covered completely, i.e. the remaining call duration was enough to cover the Count * Duration time. Let’s take an example; if we construct the formula:

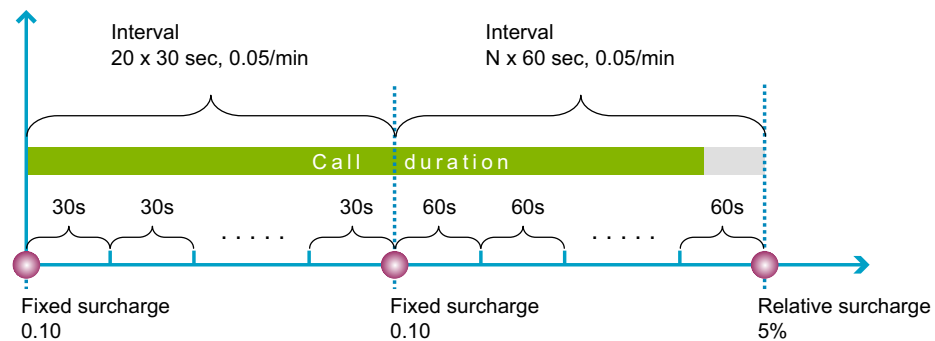
- 3 x 60 seconds, 0.10/min
- Fixed surcharge 0.05
- N x 60 seconds, 0.10/min

then, when a call is made for 1 minute and 5 seconds, the charged amount will be 0.20 – in the first interval we use two increments of 60 seconds each, with price per minute 0.10. Since this first interval is to contain three

increments, and we have used only two, the surcharge which follows the interval is not applied.

When a call is made for 4 minutes and 20 seconds, the total charged amount will be 0.55: three increments of 60 seconds each, a surcharge of 0.05, and then another two increments of 60 seconds with price per minute 0.10.

There is one exception to the surcharge application rule above: if the last formula element is a surcharge, it is always applied. This allows emulation of the “post-call surcharge” from earlier releases.



The figure above demonstrates how calls are charged. For example, the rating scheme “Charge 0.10 connection fee, then first 10 minutes should be rounded up to half a minute with price per minute 0.05. If the customer calls for more than 10 minutes, apply 0.10 fee and round the rest of the call up to 1 minute with price per minute 0.05; apply 5% post-call surcharge at the end of the call” will be translated into the following formula:

- Fixed surcharge 0.10
- Interval 20 x 30 seconds, price per minute 0.05
- Fixed surcharge 0.10
- Interval N x 60 seconds, price per minute 0.05
- Relative surcharge 5%

Most often you would like to set up a general call rating scheme (rounding, surcharges, etc.) immediately, even though you foresee that your rates per minute will change in the future. In this case, you may construct the formula as you would normally do, but instead of entering a fixed price per minute in the formula you can specify: “use the current value of the **Price_Next** parameter”. Later, when you decide to change your rates, it will be much easier to download the current rate data, alter just one column (**Price_Next**), and upload the file again, rather than change the formula in every row. Also, using the pseudo-variables First Price and Next Price is required to ensure that the correct rate is used during the off-peak period, since the system will then automatically populate the variables with the values you specified for either peak or off-

peak prices. Of course, if you entered a constant price value into the formula it will remain unchanged regardless of the off-peak period.

Using formula parameters linked to rate values has another advantage: if you have different prices or interval durations in peak/off-peak periods, the system will automatically use the one applicable.

Too short calls

Unfortunately this is a common problem: for some destinations (usually where analog lines are used) your vendor will report the call as connected once the remote phone starts ringing. So, even if no one picks up the phone, such calls will still be considered successful by your system, with a duration (for instance) of 15 seconds, and the customer will be charged for them – which will probably lead to disputes and arguments. Using the new call rating formula, you can specify that short calls to a specific destination be disregarded. So, if you set the **Do not bill calls shorter than** parameter to 20 for a certain destination, and if a call lasts less than 20 seconds, it will not be billed at all. However, if the call was made for 20 or more seconds, it will be charged for the full call duration.

Payback (reverse) charging

A rate that is marked as **reverse** contains exactly the same set of parameters as the normal rate. The total amount is calculated according to the same rules as described above – with the exception that this amount is then **credited** to the customer's account.

This allows for “awarding back” money to the customer for certain types of service (e.g. when he receives an incoming call).

Prepaid Card Billing Features

PortaBilling® offers an extremely flexible rating and authorization engine for providing prepaid calling card service. It contains the most comprehensive set of special features required for this type of service. Please see more details in the **PortaSwitch Prepaid Services Handbook** (*Prepaid Billing Features* section in the *Appendices* chapter) about the specific functionality available.

Volume-based Billing

Volume-based discounts

Sometimes you want to adjust rates based on the amount of service your customer has already used. Volume-based discounts are ideal for this. The key features of volume-based discounts include the following:

- A discount is defined per destination group. For example, if you wish to provide special rates for calls to “ex-USSR countries” you can create a destination group that will contain prefixes for Russia, Ukraine, Belorussia and others, and define a single discount rule for this destination group.
- Multiple discount steps per destination group. For example, you can define a discount rule stating: first 200 minutes free (100% discount), then up to 500 minutes at the normal rate (0% discount), then a 10% discount for every minute up to 1,000 minutes, and a 20% discount for every minute over 1,000.
- Discounts can be applied regardless of the time when the service is used. Alternatively, a different set of thresholds can be defined for peak and off-peak periods. It is also possible to define a volume discount which would only function during peak periods, but not during off-peak ones, or vice versa.
- You can use thresholds based on call time (e.g. 10% discount after 200 minutes) or on the call cost (10% discount after more than \$10 worth of calls).
- Several discount rules may be grouped into a volume discount plan. For instance, your service may include 300 free monthly minutes to US&Canada, 100 free monthly minutes to Western Europe, and a special offer of 15% off calls to India after the customer has called for more than 200 minutes. Each of the conditions above (e.g for calls to US&Canada or India) is represented by an individual discount rule. Taken all together, they will be grouped as discount plan “EasyCall”, which can then be applied to a specific account.
- For better management, you can associate a volume discount plan with a certain product. This discount plan will then be applied by default to every account using this product. You may also override the discount plan setting for a specific account, so that the system uses a specifically assigned discount plan instead of the one defined in the product.
- Discount counters (amount of currently used minutes) may be automatically reset in accordance with the customer’s billing period. This makes it very easy to implement services such as “200 free minutes per month”.

The discount is applied to every minute which exceeds the threshold. Let’s look at the following example:

- You have the discount rule “first 200 minutes – normal rate \$0.20/min, 15% discount after 200 minutes” for calls to Israel.
- During one month the customer spends 230 minutes on the phone.

In this case, the customer will be charged $200 * 0.20 + (30 * 0.20 - 15\%) = 40 + 5.1 = 45.1$. Sometimes people get confused, and will assume that the charges in this case should be $230 * 0.20 - 15\% = 39.1$, which is **incorrect**.

Discount plans may also be assigned at the customer level. In this case, they will apply to calls made by any of the customer's accounts. If both the account and the customer have discount plans assigned for the **same** destination group, PortaBilling® will combine the discounts using one of two available methods:

- When using *shared* counters (the default method), the actual discount is equal to the account's and the customer's discounts combined. For instance, if the account's discount plan provides a 30% discount, and the customer's discount plan provides a 20% discount, a 50% discount will accordingly be applied to the call. After the call is completed, the volume counters for both the account and the customer are updated; so if a 5-minute call was made, both sets of counters will increase by 5. If the sum of the two discounts is more than 100% (e.g. the account's discount plan gives a 100% discount and the customer's a 50% discount), the total discount will still be 100%.
- The alternative method is to use *exclusive* discounts. If a discount in the account's plan is defined as exclusive, and provides a 100% discount for the call, then only counters for the account will be updated, until the discount is used up. This allows you to provide stackable discounts, e.g. every account has 200 free minutes, while 500 free minutes are also available at the customer level for use by accounts which have used up their own free minutes.

Within the same discount plan you can have discounts for some destination groups based on the shared counters model, while others will be exclusive.

Threshold warnings

It is possible to configure the system so that a customer crossing a discount threshold (where a new discount level is applied) would automatically receive a notification. This could be used to notify the customer that he is now out of free minutes, or inform him that he is now eligible for a higher discount.

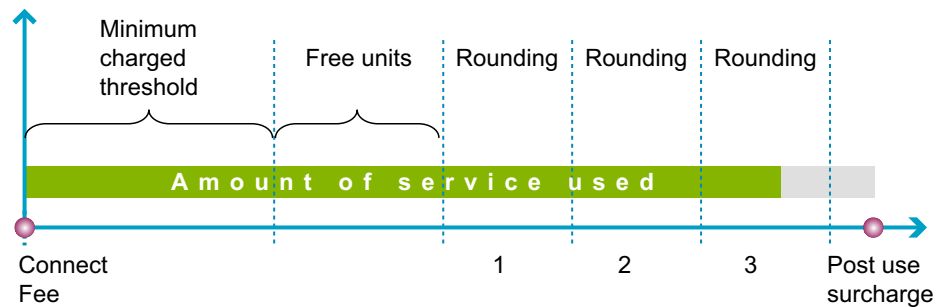
Off-peak periods and volume discounts

By default, volume discounts are applied irrespective of peak and off-peak periods, i.e. a minute is always counted as a minute. It is possible to create a volume discount plan with a different structure of discount thresholds for peak and off-peak time (and even for first and second off-peak periods). In this case, only counters for the relevant portion of the

discount plan will be modified, based on the time when the service is used.

Quantity Billing Parameters

Quantity-based services (such as messaging) are billed rather similarly to the “default” rating of calls, with various rating elements applied to the “total used service” value. In fact, if we regard time-based services as those where a quantity of seconds is used as the rating parameter, then the two are almost identical!



The picture above demonstrates how quantity-based services are charged. A **Connect Fee** is applied for all such billing events. For any event where a customer has used less than the *Minimum Threshold* units, he will be charged as if he used the *Minimum Threshold*. *Free Units* are granted after the *Minimum Threshold*, meaning that the number of units specified in this parameter will not be charged. The amount remaining after (*Minimum Threshold + Free Units*) will be rounded up to multiple **Rounding** units. Note that at this point only measurement units are being used, and these must be converted to billing units, so that the actual amounts charged will be calculated as $Price_per_unit * Units / Billing_Ratio$. After the total charged amount has been computed, the **Post Use Surcharge** is applied. The service illustrated in the figure above will be charged using the following formula:

$$Amount_Charged = (Connect_Fee + Minimum_Threshold * Unit_Price_Initial / Billing_Ratio + 3 * Rounding * Unit_Price_Next / Billing_Ratio) * (1 + Post_Use_Surcharge / 100)$$

Parameters such as **Minimum_Threshold**, **Rounding**, **Unit Price Initial** and **Unit Price Next** can be specified per destination. **Connect Fee**, **Free Units** and **Post Call Surcharge** are defined on a per-tariff basis, and so are the same for all destinations within a tariff.

To understand this better, let’s take another simple example. A data transfer service rating is defined as follows:

- The *measurement unit* is a byte, the *billing unit* is a kilobyte, and the *billing ratio* (how many measurement units make up one billing unit) is 1,024.
- The rate is defined as follows: *Minimum Threshold* – 10,240, *rounding* – 1,024, and the *initial* and *next prices* are the same – \$0.02. (Prices are defined per **billing** unit, so this is a price per **kilobyte**.)

So if a customer initiates a session and transfers 1,976 bytes, this will fall under the *Minimum Threshold*; thus he will be charged $10240 * 0.02 / 1024 = \$0.20$. If the customer transfers 17,290 bytes during the session, this results in 8 rounding increments above the *Minimum Threshold*, and so the total charged amount is $10240 * 0.02 / 1024 + 8 * 1024 * 0.02 / 1024 = \0.36 .

Overdraft Protection

The traditional authorization scheme, in which the system checks the amount of available funds at the beginning of a call, has many drawbacks. For one, the system does not react to balance changes (refunds, payments, or charges) made while the call is in progress. Since every session is authorized based on the total amount of resources available, overdrafts are possible in the case of multiple concurrent sessions and the like. The alternative – restricting the number of concurrent sessions to one – allows you to avoid overdrafts, but causes inconvenience to the end-user.

This is why PortaBilling now offers you the ability to dynamically lock the funds required to pay for a certain amount of a service before that service is used. Proper implementation of overdraft protection requires support both on the billing engine side (PortaBilling) and on the part of the network element providing the service, e.g. PortaSIP®. (See below for a description of the scenarios applicable to “legacy” switches.)

Authorization and fund locking

What is the relationship between authorization and fund locking? Authorization is simply a check as to whether the currently available funds are enough to cover a service (or how much of the service can be used with the funds currently available). Using a simple analogy, it is like taking a look in your wallet to see how much money you have left before deciding whether you have enough money to purchase a DVD or buy some chocolate bars. The problem is that such a check is done without taking into account your other attempted purchases. For example, if you have \$20 available, the system will tell you that you have enough money to buy 1 DVD (\$15) or 10 chocolate bars (\$2 each); but it cannot tell you whether you can buy a DVD and **also** buy 3 chocolate bars.

Fund locking not only checks that you have enough money for the DVD, but also immediately puts that money into a different compartment of

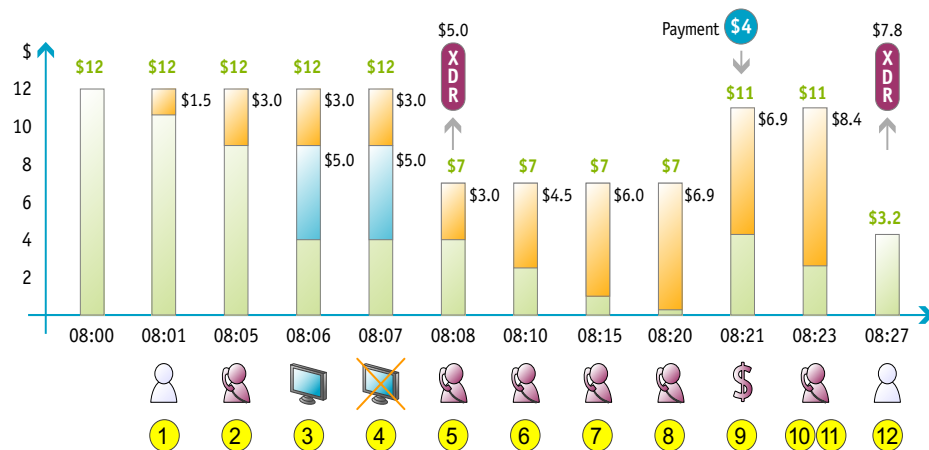
your wallet. Now when you check to see how many chocolate bars you can buy, you will only be working with the remaining portion of the money. This will spare you an awkward situation at the cash register.

Dynamic authorization

The simplest overdraft protection scenario is “resource locking”. This is done when an application knows the amount of funds required beforehand. So right at the start the application sends a request to validate the availability of these funds and, if this is successful, it locks them. Then, when the purchase has been completed, a “charge” event is sent to PortaBilling. One example of such a service is pay-per-view content distribution. After a customer clicks on a “Get the movie” link, the system authorizes and locks the funds to cover the cost of the movie, in order to ensure that the customer has the right permissions and sufficient funds. The customer is then shown a confirmation screen. If he approves the purchase, the money is deducted from his account

Dynamic re-authorization is a more complex process, which is used when it is not possible to tell how much a customer will spend in a particular session, so that additional funds have to be locked “on the go”. When a session (e.g. a voice call) starts or is in progress, the system “reserves” a small portion of the available funds to cover the next time interval (e.g. 5 minutes). As the end of this interval comes near, the system attempts to reserve funds to cover the next interval, and so on. If no remaining funds are available, the call is disconnected. So if a customer sends an SMS message during a phone call (decreasing the available funds), or makes a credit card payment (increasing the available funds), this will immediately affect the call – either it will be disconnected sooner, or the maximum allowed call time will increase.

Let’s see how dynamic re-authorization works in a specific example, where an account is concurrently using multiple services and making a payment while the session is still in progress.



- Initially, the account has \$12 available.
- The user starts a phone call to a destination with a price of \$0.30 per minute. The PortaSIP server requests call authorization for 5 minutes. At this point the user's balance is still unchanged, but funds covering a 5-minute phone call (\$1.50) are locked (1).
- While the call is in progress, at some time before the end of the "approved" period (for our example, we'll assume this is one minute), the PortaSIP server will attempt to perform re-authorization.
- PortaSIP sends a re-authorization request to billing. PortaBilling verifies that the account has enough funds for another 5 minutes. The amount locked is now \$3.00 (2).
- While still on the phone, the user decides to purchase a pay-per-view movie. An authorization request arrives from the IPTV platform, and the funds needed to pay for the movie (\$5.00) are locked in addition to the previous \$3.00 (3).
- The user now attempts to purchase another movie (\$5.00), but this request is rejected, since he only has \$4.00 left (4).
- The purchase of the first movie is completed, and the balance decreases to \$7.00 (5).
- Since the call is still in progress, and the end of the "authorized" time is approaching, PortaSIP sends another authorization request, and so funds for another 5 minutes are locked (a total of \$4.50 is now locked) (6).
- The user keeps talking, and so 5 minutes later yet another authorization is sent. The total amount of locked funds is now \$6.00 (7).
- Soon another authorization is sent, but now there are only enough funds for 3 minutes. \$6.90 is now locked (8).
- The user realizes that he is running out of money, but wants to continue the phone call, and so he promptly makes an online payment with his credit card. \$4.00 is deposited to his account (9).
- Since the last re-authorization provided less time than requested (3 minutes instead of 5), PortaSIP determines that the user is out of money, and normally it would disconnect the call. However, it makes one last re-authorization attempt, one minute before the call is to be disconnected (10).
- Since the user now has available funds, another 5 minutes of call time are authorized. The total amount of locked funds is now \$8.40 (11).
- Finally, the user disconnects, and is charged \$7.80 for the full call duration of 26 minutes. Thus his balance is now \$3.20 (12).

Authorization chunk

Two situations may lead to an excessive locking of funds:

- Attempting to use an “expensive” service – For instance, let’s assume that PortaSIP® is configured to lock sufficient funds for a 15-minute conversation at the start of a call. This is fine for calls to US or Europe (in which case just 20-40 cents would be locked). But if a customer with a fairly low balance calls an expensive destination (e.g. Somalia – \$1.00/min), he may find that all of his funds are suddenly locked up, and will not be able to use another service in the meantime.
- Old-style applications that do not support re-authorization – If an authorization request comes from a gateway or switch which does not support dynamic fund locking, then all the available funds will be locked while that session is in progress.

To prevent locking too many or all available funds, the administrator can configure the product’s **Authorization Chunk** parameter.

Add Accessibility to Product 'EasyCall'			
Product Currency		USD	
Service	Voice Calls	Node	PortaSIP
Access Code		Info Digits	ANY
Authorization Chunk	3.00	USD	
Default Tariff *	EasyCall		

Then each request will lock up no more than the amount specified there. Taking our example of a call to Somalia, if we set the authorization chunk to \$3.00, the billing engine will respond to PortaSIP’s request to lock funds sufficient for 15 minutes by locking up only \$3.00 (3 minutes). Thus PortaSIP® will perform authorization every three minutes.

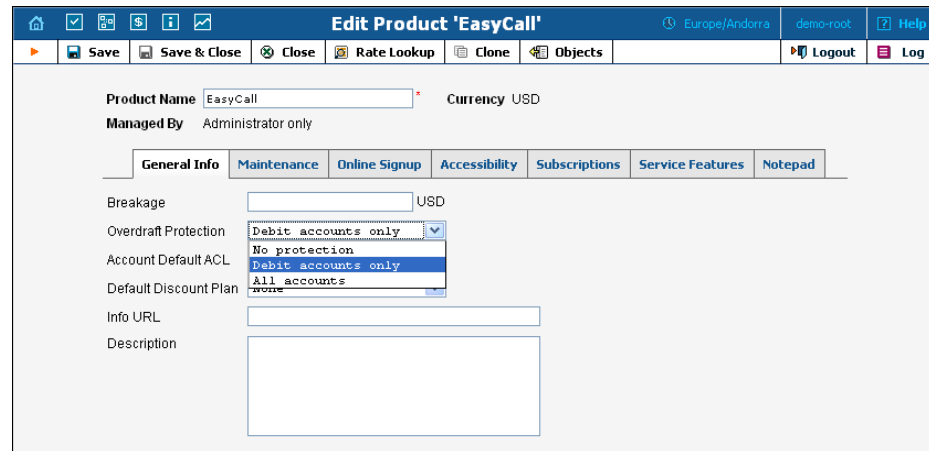
Legacy equipment and backward compatibility

In order to fully use the overdraft protection capabilities, the node providing the service (gateway, switch, etc.) must send the proper attributes in the RADIUS request to PortaBilling. During the initial authorization request, it will inform PortaBilling of the desired interval for fund locking (and the minimum possible interval for re-authorization), and will then issue re-authorization requests periodically.

Currently, real-time overdraft protection is fully supported by the PortaSIP® call processing engine.

What happens when a node that is not capable of dynamic re-authorization sends a “standard” RADIUS authorization request? Also, how does fund locking affect the previously available “fraud protection” feature in PortaBilling (where a debit account is allowed to make only one simultaneous call)?

Answer: the product configuration features an **Overdraft Protection** setting, which allows you to fine-tune how fund locking is applied:



The screenshot shows the 'Edit Product 'EasyCall'' interface. The 'Overdraft Protection' dropdown menu is open, displaying the following options: 'Debit accounts only', 'No protection', 'Debit accounts only', 'All accounts', and 'none'. The 'General Info' tab is active, and other tabs include 'Maintenance', 'Online Signup', 'Accessibility', 'Subscriptions', 'Service Features', and 'Notepad'. The 'Product Name' is 'EasyCall' and the 'Currency' is 'USD'. The 'Managed By' field is set to 'Administrator only'.

- **None** means that there are effectively no “locked” funds. When a locking of funds is requested, these locks will be done separately for each session, and will not affect other sessions. So there can be several concurrent sessions, and each can potentially use up all the available funds. This method thus provides transparency for the application using overdraft protection, although in reality it does not provide greater security than a simple authorization request. It is analogous to “Fraud Protection Off” mode in previous releases. For obvious reasons, it is not recommended for general use.
- **Debit accounts only** – In this case, no fund locking will be done for credit accounts, while for debit accounts the requested amount will be locked. When an authorization request is received from a “legacy” application for a debit account, the billing engine will lock either all of the available funds on that account, or only the Authorization Chunk amount. This is identical to PortaBilling’s default behavior in Maintenance Release 18 and older.
- **All accounts** – This is used where fund locking is done for all account types. For “overdraft protection-aware” applications, the amount of funds locked is the smaller of either the amount needed to cover the requested service amount, or the **Authorization Chunk**. For legacy authorization requests, it is either all of the available funds, or the **Authorization Chunk**. This is probably the most secure mode, but it may be too restrictive in the case of legacy equipment.

xDR Recalculation

Mistakes can happen – someone sends you the wrong pricelist, or your administrator simply clicks the wrong button, resulting in incorrect charges in the database. How this can be corrected before an angry customer calls you?

PortaBilling® provides two methods for fixing operator errors:

- RADIUS re-feed
- xDR re-rating

RADIUS Re-feed

Every accounting request received by the billing engine is stored in a separate file on the master server. A special command-line utility supplied with PortaBilling® can be used to extract all RADIUS requests for a specific time interval (starting from the time when the error was made until the time it was fixed) and resend them to the RADIUS server. Each request is tagged, so that a special processing mode is invoked.

- Any existing xDRs for this session are erased from the database, and balances are rolled back (as if the previous charge attempt had never happened). This also includes turning back any volume discount counters that were modified.
- The session is charged according to the new (correct) billing configuration.

Re-feed is a very powerful tool, and can fix any type of billing error (including missing connection to vendor, incorrect product accessibility, or wrong rate). However, it must be used with caution, as it involves processing large volumes of data. Customers are advised to contact the PortaOne support team, who can provide further assistance in running the re-feed.

xDR Re-rating

This is a lightweight and easy-to-use alternative to re-feed, designed to fix the most common problem: incorrect pricing information entered into a tariff. It is operated completely from the PortaBilling® web interface. The administrator specifies the “original” (incorrect) tariff and the correct tariff, and then narrows the set of xDRs to be processed by time interval and specific customer.

The re-rating task starts in the background, and during its execution:

- The script locates all xDRs produced according to the incorrect tariff (and matching the time interval and customer criteria).
- For every xDR, the script calculates the charge according to the new tariff. If the charges are identical, no further action is taken with the given xDR.

- The xDRs are then updated with the correct charged amount, and the balance is adjusted by the difference between the new and old charged amount.

Re-rating and volume discount counters

If volume discounts are used when calculating call charges, it is no longer possible to treat such calls separately from others, since the way one call is charged affects all other calls made subsequently. For instance, if a call is charged \$5, this is the amount added to the volume discount counter. If the charged amount is then changed to \$1 during re-rating, this will affect all other calls in the same destination group, since they can now be charged at a different discount rate.

To overcome possible confusion when volume discount counters are involved in re-rating, this process should always be run from a specific moment in the past (when the error occurred) to the present moment. In this case, all discount counters will be rolled back before recalculation actually starts, and then updated with each re-rated call.

NOTE: Re-rating with volume discount counters only works with XDRs generated following an upgrade to the MR16 version, since older versions of XDR data do not have the full information required to roll back volume counters.

Subscriptions

The **Subscriptions** module allows you to charge your customers periodic fees for using the service. If your advertisement states something like: “only \$9.99 per month, and just \$0.99 extra per month for Voicemail service”, two subscription plans are involved here. When you define a subscription plan, you define various parameters that specify what effect the plan will have on the customer. Every subscription plan includes the following:

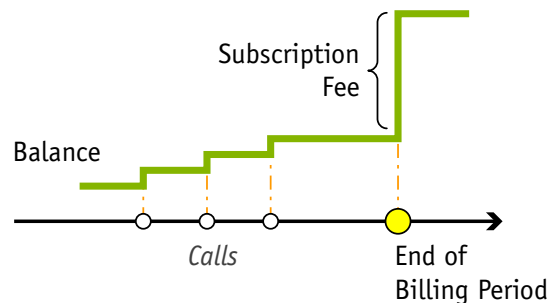
- Name and description – Used by your administrators to better manage various subscription plans.
- Invoice line description – The subscription name your customers will see on the invoice.
- Activation mode – Specifies when the subscription is active and charges start to be assessed. Typically, a subscription is activated immediately upon its creation (or on the start date, if a future start date has been specified). PortaBilling® supports an additional mode of subscription activation based on the account’s first use. This allows you to avoid problems when there is a delay between the time a customer signs up for a service and the time he is actually able to use it (e.g. he uses online signup to purchase a residential SIP service, but his IP phone will only be delivered by FedEx five days later).

- Activation fee – The amount to be charged when the subscription becomes active.
- Minimum subscription period and early cancellation penalty – It is common practice to lock a customer’s contract for a certain period (e.g. if you provide the customer with a free IP phone, you want to make sure he keeps paying monthly fees until he has paid back the cost of the phone). If the subscription service is canceled earlier than the interval specified, the customer will be charged a cancellation penalty.
- Rounding – To avoid unusual subscription charges like \$1.26789, you can specify a rounding pattern for them (in a similar way as you have done with rating in tariffs). For example, the rounding pattern XXXXX.XX000 will round up all subscription charges to whole cents.
- Of course, every subscription also includes a definition of periodic charges applied while the customer has an active subscription; see below for more details.

Applying Subscription Charges

Traditional method

This is a very common and widely popular method. When a customer’s billing period is closed, PortaBilling® calculates the applicable charges for all subscriptions which were active during that period, and applies these to the customer’s balance.

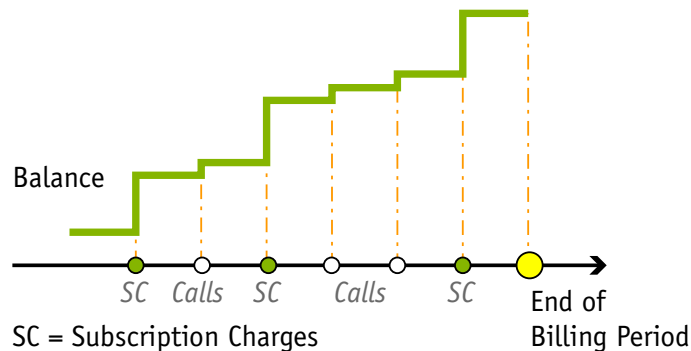


As a result, at the moment the billing period is closed there may be a significant increase in the customer’s balance.

Progressive method

While the situation described above is acceptable for postpaid customers with a large credit limit, or if the customer has a credit card which may be charged automatically when the billing period is closed, it may create certain problems for prepaid customers. If such a customer does not have sufficient available funds at the end of the billing period, his balance will exceed the credit limit, and his outgoing calls may be blocked.

PortaBilling® provides a solution to this problem, allowing you to conveniently offer prepaid VoIP services in combination with subscription fees.



To avoid the unpleasant situation of a sizeable charge at the end of the billing period, PortaBilling®'s progressive charges are applied continuously, so that, at any given moment in time, the amount covering the interval from the beginning of the billing period until the current day has already been charged to the customer. For example, if a customer's monthly fee is 9.99, on the first day of the month he will be charged $9.99 / 30 = 0.33$. On the next day the charge will be 0.66, and so on until the end of the month, when the charge will equal the full amount of 9.99. Thus the customer is effectively charged a small portion of the total fee every day. His balance grows slowly, giving him enough time to react and deposit more money into his account.

Subscriptions charged In advance

This mode of charging subscription fees ensures that, at the end of the billing period, the customer is charged until the end of the Nth full billing period following the current one. So if we charge one month in advance and the current monthly billing period ends on April 10th, the customer will be charged until May 10th, and his invoice produced on April 11th will contain charges for calls made prior to April 10th and subscription charges covering the period from April 11th to May 10th.

If a subscription which is charged in advance is not active during the whole billing period, it is pro-rated in the same way as post-charged subscriptions are. For instance, if a customer with a monthly billing period activates a "1 month in advance" subscription in the middle of his billing period (e.g. on March 17th), the invoice issued to him on April 1st will include subscription fees from March 17th to April 30th (i.e. the incomplete portion of the current billing period and the whole next billing period).

If a customer terminates a subscription prior to the end date of a period that has already been charged (e.g. the customer in the example above closes his account on April 20th), then a refund will be issued for the "unused" subscription time when the billing period is closed.

This improves cash-flow for a service provider, and minimizes the potential loss when a customer leaves without paying his last bill.

Periodic Subscription Fees

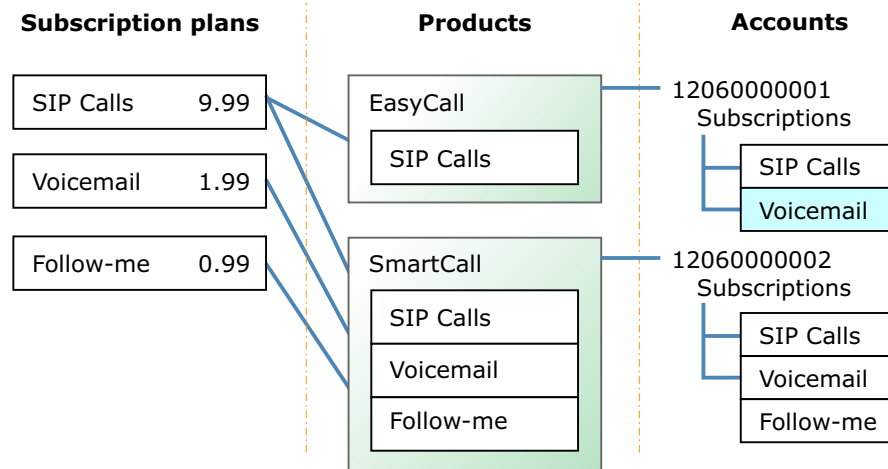
As mentioned above, periodic fees are charges applied continually throughout a subscription's lifetime. Since your customers may use different billing periods, PortaBilling® gives you the flexibility to assign subscription costs individually for each period. For instance, if your monthly rate is 19.99, you will not want to offer a weekly rate of 4.99, since the increased maintenance required by customers with a short billing period justifies increased rates.. PortaBilling® allows you, for instance, to define your monthly rate as 19.99, bi-weekly as 10.99, weekly as 6.99 and daily as 1.99 for the same subscription plan. PortaBilling® will automatically use the correct base value according to the customer's billing period.

Promotional periods

Yet another common business practice is offering special rates for an initial period following signup (e.g. "Only 9.99/month!", while the disclaimer states: "For the first six months only, after which the standard rate of 29.99/month applies."). PortaBilling® allows you to define an unlimited number of subscription periods, with different subscription fees for each. For example, you could create a subscription which offers free service for the first three months, a rate of 19.99 for the next nine months, and 12.99 thereafter; or something even more complex.

Applying Subscriptions

Since a product defines the way you provide a service to the end-user, and subscriptions define the charges for this service, it is obvious that the two must be interconnected. There is now a new property of products – the subscription list – that defines which subscription plans are to be included in a given product. For instance, your EasyCall product may only include one subscription plan, e.g. "SIP calling – 9.99", while the SmartCall product might include "SIP calling – 9.99", "Voicemail – 1.99" and "Follow-me – 0.99". A SmartCall customer will be charged 12.97 in total, but there will be three separate invoice lines, one for each subscription plan. Thus, when an account is assigned the SmartCall product, three subscription records are created for it (SIP calling, Voicemail and Follow-me), while only one subscription record is created for an account with the EasyCall product. These **subscription records** (not to be confused with **subscription plans**) reflect the subscription's actual status with regard to this account.



Subscriptions which an account receives as part of its product are called mandatory subscriptions, because it is not possible to remove or cancel these subscriptions so long as the account uses the product. You may, however, add some optional subscriptions to an account (e.g. if an EasyCall product user wishes to receive the Voicemail service). These optional subscriptions can be added or canceled as you wish.

You can also assign subscriptions to a specific customer. Here the procedure is exactly the same as with an account. Since a customer cannot have a product assigned directly to him, there are no mandatory subscriptions in this case, i.e. all of a customer's subscriptions are optional.

Every subscription has parameters such as activation date, cancellation date and (most importantly) the “billed to” date, which defines the period for which subscription fees have already been charged.

Activating a subscription

By default, a subscription is automatically activated when its start time arrives, with one exception. Namely, in the subscription plan properties you may specify that the subscription will become active only when the account is used for the first time. In this case, the subscription is activated either on the start date or the account's date of first use, whichever comes later.

Canceling (closing) a subscription

A subscription which is not yet active can simply be deleted (for instance, a customer has signed up for a new service beginning next month, but then changes his mind). A subscription which is already active cannot be deleted; when the subscription is canceled, the cancellation date is entered as the end date of the subscription. Typically, the cancellation date will be the day on which cancellation took place, but you can also schedule

cancellation for a future date (e.g. a customer wishes to use the service until the end of the present month). The record of the canceled subscription allows PortaBilling® to correctly charge for closed subscriptions. For example, if a customer subscribed to the service on the 3rd and cancelled it on the 7th, he will still be charged for five days of service.

Subscription Discounts

Discounts on subscriptions

Every subscription (assigned to an account or a customer) can have a custom discount rate, with the actual charged amount adjusted accordingly. So if a subscription plan defines the periodic fee as \$10.00, and this particular subscription is assigned a 20% discount rate, a charge of only \$8.00 will be produced. Use 0 (zero) discount rate to specify “no discount” and ensure that charges are made according to the subscription plan definition. If a subscription has an empty discount rate, the customer default rate is used (see below).

Customer discounts

To simplify management of different discount tiers (e.g. all your “silver” customers get 10% off the monthly fee for any of their subscriptions) you can assign a discount rate directly to a customer. So instead of creating multiple subscription plans with different periodic fee values, you will have a single set of subscription plans. You can then simply assign discount rates (10%, 20%, etc.) to customers. Any subscription (either directly applied to the customer or associated with one of the customer’s accounts) which does not have an explicitly assigned discount rate will then be charged using this discount.

Product Modifications

Changing the subscription plan definition of a product

If you modify the list of subscription plans for a given product, this will not affect existing accounts with this product. This basically allows you to sell the same product over an extended period of time, simply adjusting the subscription plans within the product according to current promotional offers, without affecting any old customers.

If you wish to update subscriptions for existing accounts with this product, you must explicitly request it by clicking the **Reapply Subscriptions** button.

Changing an account's product and subscriptions

A change of product for an existing account, e.g. from product A to B, will have the following effect on this account’s subscriptions:

- Mandatory subscriptions present in both A and B will remain unchanged.
- Mandatory subscriptions which were part of product A (old), but not part of product B (new) will be closed (with the current date as finish date).
- Mandatory subscriptions under product B which the account does not yet have will be created as new ones (with the next day as start date).
- Optional subscriptions will remain unchanged.

Applying Subscription Fees

When a customer's billing period is closed, PortaBilling® calculates charges for any subscriptions that were active during this period. For each subscription, three types of charges may be made:

- Activation fee – If the subscription plan has an activation fee defined, and if the subscription was started during this period.
- Cancellation fee – If the subscription plan has a minimum subscription period and early cancellation fee defined, and if the subscription was canceled during this period (where the total subscription duration is less than the minimum subscription time).
- Subscription fee – A periodic subscription fee, pro-rated to the actual duration of the service used; please consider the following examples, all of which apply to the “SIP calls” subscription with a monthly periodic fee of \$9.99:
 - Customer A, with a monthly billing period, activated the subscription on April 12th. On May 1st he will be charged \$6.33, because he used the service for 19 days (12..30) and $(19/30) * 9.99 = 6.33$
 - Customer B, with a monthly billing period, activated the subscription on April 12th, and canceled it on the 25th. On May 1st he will be charged \$4.66, because he used the service for 14 days (12..25) and $(14/30) * 9.99 = 4.66$
 - Customer C, with a monthly billing period, canceled his subscription on April 23rd. On May 1st he will be charged \$7.66, because he used the service for 23 days (1..23) and $(23/30) * 9.99 = 7.66$

All these types of charges will be reflected on the invoice as separate invoice lines.

Incompletely used promotional periods

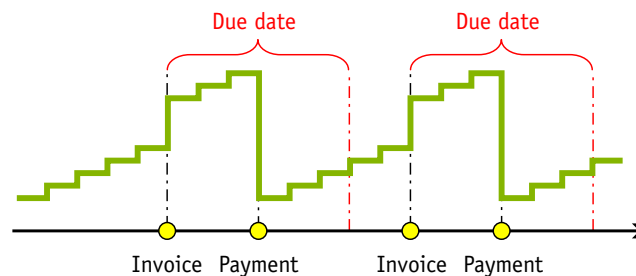
If you offer promotional rates for X billing periods, then every billing period in which the service was used counts towards it, regardless of the amount of days used in a particular billing period. Thus, if you offer a promotional period for the first three months, and customer subscribes on a July 15th, the second half of July (16 days) and the two months

following it (August and September) will be charged according to the promotional rate. Starting from the fourth month (October) the default rate will be used.

In order to avoid potential misunderstandings when a customer expects a longer promotional period, the best practice is to use an anniversary billing cycle, as this eliminates the problem of “incompletely used” promotional periods.

Invoicing

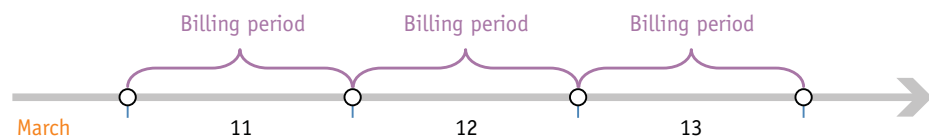
At the end of the billing period, PortaBilling® can produce an invoice for your customers. An invoice reflects all the transactions (calls, payments, refunds, subscription charges, and so on) which occurred during this period, and serves as the primary record of services provided to a specific customer, as well as his current status.



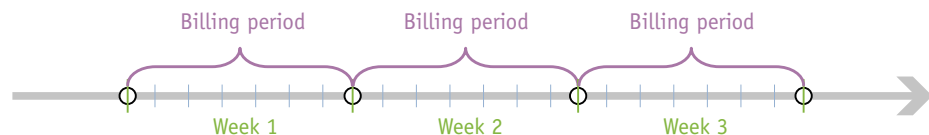
Billing Periods

A billing period defines how often invoices / statements will be generated and when exactly they are to be produced. PortaBilling® supports the following billing periods

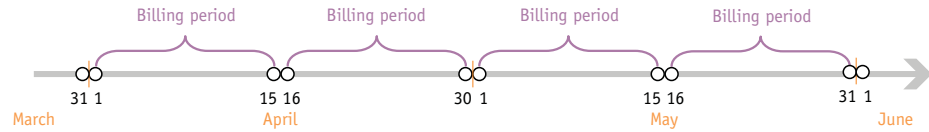
- Daily – Covers a 24-hour period.



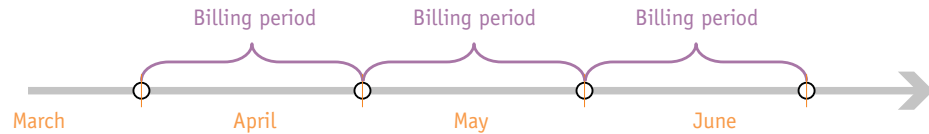
- Weekly – Covers a 7-day period (Monday through Sunday).



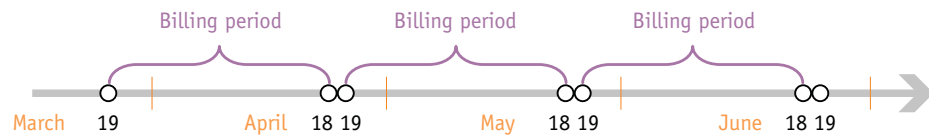
- Bi-weekly – Covers a period from the 1st to the 15th or from the 16th to the last day of the month.



- Monthly – Covers the period from the 1st of the month to the last day of the month.

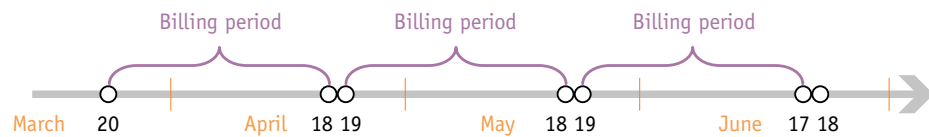


- Monthly (anniversary) – Covers the period from the Nth day of the month to the day before the Nth day of the following month. N is the day of the month when the customer was created; therefore, if a customer was created on March 19th, his invoices will always cover the period from the 19th of the current month to the 18th of the following month.



To avoid complications for customers who were created on the 29th, 30th or 31st day of the month, their first billing period will cover the time until the 28th day of the following month, and thereafter will always cover the period from the 28th until the 28th. For instance, if a customer was created on March 30th, his first invoice will cover the period from that day until April 28th, while his next invoice will cover the period from April 28th until May 28th.

- 30 days – Every billing period is exactly 30 days, so if a customer was created on March 20th, his first invoice will cover the period from March 20th to April 18th, the second invoice will cover the period from April 19th to May 18th, and so on. This invoicing method allows you to make subscription fees more straightforward compared to regular monthly billing, where the same monthly fee is applied to longer (e.g. March) as well as shorter periods (e.g. February or April).



Every billing period is adjusted to the respective customer's (or vendor's) time zone. So, in the case of a customer with the Los Angeles time zone

and a weekly billing cycle, the billing period will start at midnight on Monday Los Angeles time, while for a customer with a weekly billing cycle and the Singapore time zone it will start at midnight on Monday Singapore time. Thus while both invoices will cover 7 days (168 hours), they will actually refer to different intervals of time, with a 15-hour difference.

Closing a Billing Period

Or in other words: when will invoices actually appear in the system?

By default, invoices/statements are generated on the day after the last day of the billing period, e.g. invoices for customers with a weekly billing cycle are generated on Monday, while invoices for customers with a monthly billing cycle are generated on the first day of the month. A transaction is included in a certain billing period if it **started** during this period; this means that calls are considered as falling within a certain period according to their connect time. For example, if a call starts at 23:55 (11:55pm) on March 31st and finishes at 00:43 (12:43am) on April 1st, then this call belongs to the March billing period.

For this reason, a billing period cannot be closed the next day at midnight sharp; there might be calls in progress which started just a few minutes ago, and which should still be included on the current invoice. PortaBilling® waits a sufficient amount of time before closing a billing period, to ensure that all calls have been completed. By default, this interval is six hours, but it can be changed in the `porta-admin.conf` configuration file.

Also note that since statements are generated in the time zone of the customer/vendor, the billing period is closed in that time zone. Thus if your system time zone is Singapore, you cannot expect to see invoices for your US customers on Monday morning, since it is still Sunday evening then in the US. Finally, in order to provide optimal system response time for your online users, PortaBilling® only performs resource-intensive calculations (such as creating statistics/invoices) during the period specified in your configuration as “off-peak”. So, in the example above, if your “statistic calculation” hours are defined as 2:00am to 7:00am, you will receive invoices for US customers on Tuesday morning.

Invoice Templates

You can design multiple invoice templates, so that each template has its own layout, language, logos/pictures, and the like. Every customer will be assigned a specific invoice template, according to which PortaBilling® will create a PDF file that can be emailed to the customer and/or downloaded from the PortaBilling® web interface (by both the administrator and the

customer himself). It is also possible to use a special “Do not create invoice” option for a customer. In this case, PortaBilling® will only produce CDR summaries, but no actual invoices will be generated.

Please consult the [PortaBilling Templates Guide](#) for more information about creating and managing invoice templates.

Invoice Parameters

Every invoice contains global invoice data, which is stored as part of the invoice record in the database:

- Invoice number – unique identifier of an invoice
- From date – start date of the billing period
- To date – end date of the billing period
- Invoice date – date when the invoice was generated
- Due date – date by which payment should be received
- Payment terms – description of payment terms (e.g. “due on receipt”)
- Invoice total – sum of all charges in this period minus credits/refunds
- Invoice amount due – amount the customer is supposed to pay you (see below for a detailed explanation of invoice balances)
- Invoice status – **Open** means that the invoice has been generated, but has not yet been delivered to or viewed by the customer, so it can potentially be modified. **Closed** means that the invoice has already been sent to the customer by email or downloaded by him from the web interface.
- Invoice payment status – this specifies the following:
 - **Do Not Pay** – invoice amount is 0, no payment is required
 - **Unpaid** – no payment has been received yet
 - **Partially Paid** – payment has been received, but in an amount less than the amount due
 - **Paid** – invoice has been paid in full
 - **Overdue** – invoice is unpaid and past due
 - **N/A** – payment status is not applicable to this invoice.

Invoices also contain certain data extracted from other PortaBilling® objects, which are included in the invoice’s PDF version:

- Information about the customer (invoice recipient)
- Information about the invoice issuer (your company or a reseller, if the invoice was issued on behalf of a reseller)
- Information about calls and other transactions included in the invoice

Invoice Balance

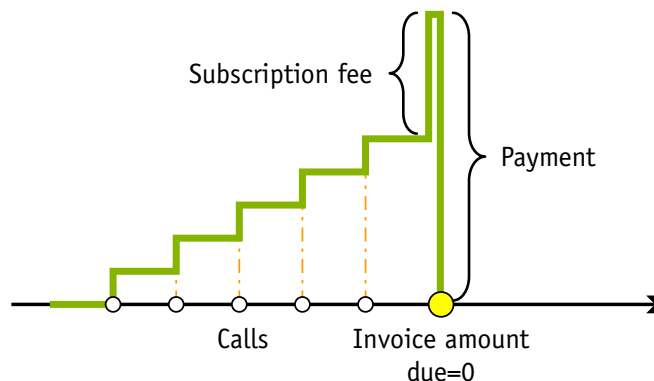
PortaBilling® provides two methods for computing the invoice balance (amount due):

- Simple – In this case, the invoice’s amount due is equal to the invoice total, and is calculated as the sum of all charges during the given period (no previous payments are taken into consideration). This is an optimal method for integration with an external bookkeeping system, where you keep track of your incoming payments via accounting software, and not PortaBilling®.
- Balance-aware – The invoice total is calculated as the sum of all charges during the given period (both call and non-call related), minus the sum of all credits/refunds. The invoice’s amount due is calculated as: $\text{previous_balance} + \text{invoice_total} - \text{payments}$. This allows you to “carry over” a balance in the case of partial payments. For example, suppose a customer’s March invoice was for \$40. If he makes a payment of \$30 on April 10th, makes calls for \$25 during April, and is also issued a \$3 refund, then his April invoice will have an invoice total of $\$25 - \$3 = \$22$, while the invoice’s amount due will be $\$40 + \$22 - \$30 = \32 .

By default, PortaBilling® uses the balance-aware invoice generation method.

Charging the Invoice Balance

To simplify payment collection and improve cash-flow, PortaBilling® can charge a customer’s credit card before closing an invoice. So if, as in the example above, the invoice’s amount due was initially \$32, the customer’s credit card will be charged \$32, with payment entered to the April billing period. As a result, the customer’s invoice will be created with a zero amount due.



This type of periodic payment can be activated on a per-customer basis (using the Payment Info tab) and used in conjunction with other types of periodic payments (e.g. balance-driven).

Payment Status

Every time a payment is recorded in the system (this could be a periodic payment, an online payment by a customer, or a payment entered manually by the administrator), in addition to modifying a customer's balance it will also be applied to one of his unpaid invoices. If the amount so applied equals the invoice's amount due, the invoice becomes "paid", while if the payment is less than the amount due, the invoice becomes "partially paid". Payments are applied to an invoice cumulatively; thus if an invoice is for \$30, and the customer makes three payments of \$10, \$13 and \$17, following this last payment the invoice will be "paid". If a customer has several unpaid invoices, the payment will be applied to the oldest one. If a payment exceeds the total amount of all unpaid invoices, the remaining sum will be assigned to the special customer property "unallocated payments", and applied to future invoices. For instance, suppose a customer receives his weekly invoice, with an amount due of \$8.99. Since he plans to leave for a three-week vacation, he sends in a payment of \$36. This entire amount is applied to the customer's balance, so that \$8.99 will cover the existing invoice, while \$27.01 will remain in "unallocated payments". When his next several invoices are created, they will show an amount due of zero and the status "paid in full".

NOTE: Unallocated payments do not represent a "cash reserve". When a payment is made, the amount is immediately applied to the customer's balance. Unallocated payments merely show that the customer "overpaid" you sometime in the past, and are used to correct the paid/unpaid status of future invoices.

You can see the customer's current "unallocated payment" status on the **Payment Info** tab in the customer info.

Collection Policy

Customer class allows you to define a policy for automated payment collection.

Important dates

- Issue date defines the moment when an invoice was issued; all other dates are generated based upon it.
- Invoice grace period indicates within how many days following the invoice issue date a payment must be received. The invoice due date is calculated as `invoice_issue_date + grace period`. The grace period can be zero, in which case the invoice is considered to be due upon receipt.
- Suspension time defines within how many days the customer will be suspended after an invoice becomes overdue. Suspension means that, while the customer's information will remain intact,

he will not be able to use certain services, most notably sending or receiving calls.

- Closing time defines within how many days a customer account will be closed after an invoice becomes overdue.

Let's take a look at the following example. Customer A has a grace period of 21 days, a suspension time of 14 days, and a closing time of 21 days. A's invoice was generated on May 1st, so that the invoice due date is May 22nd. If A does not pay by that date, the invoice will become overdue; 14 days (2 weeks) after that – on June 5th – his account will be suspended; and on June 12th – 21 days after the due date, or one week after the suspension date – his account will be permanently closed.

Collection threshold

In case a customer has not actively used the service and his invoice amount is very low, (e.g. less than \$1) it does not really make sense to follow the normal collection process and request payment for that amount.

So if the amount due on a new invoice is lower than the specified threshold – no payment is yet required. If no payment is made, the balance is applied to the next invoice, etc. until the amount due on a new invoice crosses that threshold. When full payment is made for these outstanding invoices, it will be applied to all the open invoices and then they will be considered paid.

Notifications regarding payment due

Since it may happen that a customer has lost or overlooked the original invoice, PortaBilling® can send automated notifications to a customer several times before the invoice due date, stating that payment has still not been received. You can configure any set of days, e.g. "10,7,1" will send notifications 10 days, 7 days and 1 day prior to the invoice due date.

Overdue invoice notifications

When an invoice is past its due date, PortaBilling® will continue notification attempts, but in this case using a different notification text. Here again, you have complete freedom in configuring this notification policy: for instance, "0,7,14" will send an alert to the customer on the due date, and then 7 and 14 days later.

Re-collection attempts

It could be that the initial attempt to charge a customer's credit card has failed due to a temporary problem (e.g. he exceeded his credit limit, having recently made a very expensive purchase, but has now paid his credit card bill). In such a case, it would be useful to try charging his card

sometime later. However, since the merchant bank will usually charge you for every failed credit card transaction, this should not be done too often.

In the customer class definition you can specify when re-collect attempts should be made, e.g. “0,3,7” means that PortaBilling® will attempt to charge the customer’s credit card on the due date, 3 days after the due date, and 7 days after the due date.

Suspension warning time

As a last resort to prevent service interruption, PortaBilling® will send another alert to the customer prior to service suspension. This parameter may also be configured in the customer class definition.

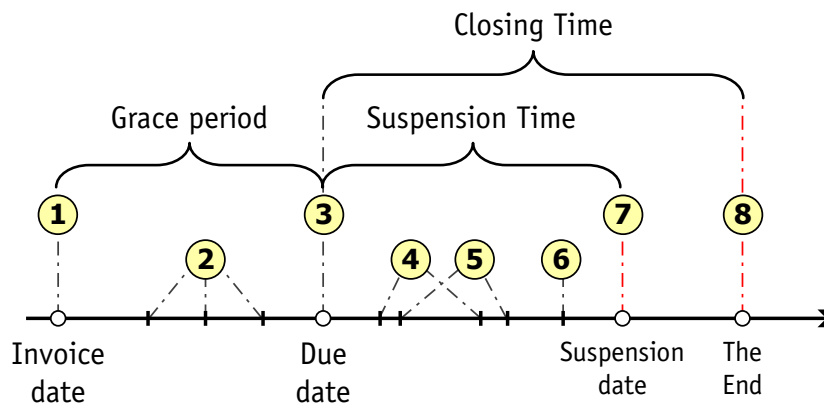
Customer account during collection

When a customer has an unpaid or overdue invoice, all call services are rendered as usual, subscription fees are charged for the current period, and new invoices are generated (for instance, if your customer has net payment terms of 60 days, he may have two other (newer) invoices generated by the time his invoice becomes overdue).

When a customer has suspended status, his call services are blocked, but subscription fees are still applied (since his account is being provisioned, e.g. his voicemail is kept on the server).

When the customer’s account is closed, no further activities, be it calls, subscription fees, self-care access, or any other, are available for his account.

Collection process



- At the end of the billing period an invoice is generated (1).

- If the invoice has a positive amount due, it is considered unpaid. When the invoice status is unpaid, a customer may be reminded of the approaching due date (2).
- On the due date, the unpaid invoice becomes **overdue** (3).
- Several collection attempts may be made (attempts to charge the customer's credit card on file for the amount due) (4).
- The customer may be reminded that the invoice is overdue (5) and that service may soon be suspended (6).
- On the suspension date, the customer's status changes to suspended (7), which automatically blocks his access to services.
- If the payment issue is not resolved, the customer's account is closed on the closing date (8).

In some cases, after the customer is suspended, and thus finally realizes that there is an unpaid invoice (!), he needs some extra time to submit payment. In this case, his suspended status may be temporary lifted by the administrator. The administrator will revert the customer's status to normal and specify the date until which suspension is postponed. If the invoice is not paid in full by that date, the customer will be automatically suspended again.

Void Invoices

It sometimes happens that an error is detected after an invoice has been generated and delivered to the customer. A new invoice must be produced, but the old one must be kept for audit purposes. The void invoice operation marks the invoice as canceled (this will also be visible in the PDF file), and then a new invoice is automatically produced.

Invoice Recalculation

This process voids an existing invoice and generates a new one in its place.

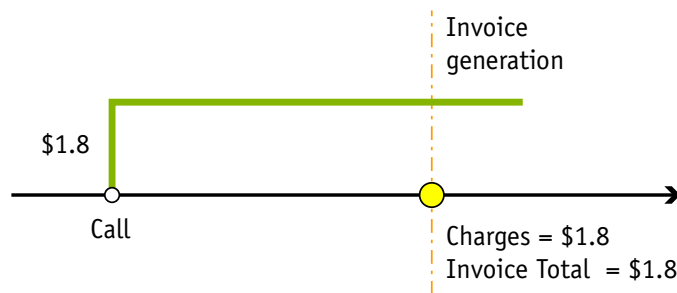
Processing Taxes

There are two methods for calculating taxes: inclusive and exclusive. Inclusive means that the rate in the price list is defined with all the applicable taxes included; so then a single xDR is produced, and the total amount in the xDR includes both charges for services and the taxes applied to them. In the exclusive method, the rate is defined without any taxes, and the total amount in the xDR includes only charges for services. Taxes are calculated later on, and added as separate xDRs. With PortaBilling®100 you can use both methods simultaneously to produce the same final result. In the examples given below, you will notice that, although different taxation methods are used, the information

on the invoice looks the same. Deciding which method to use for a specific group of customers depends on various aspects of the two methods, which are explained in detail below.

Inclusive Rate (single xDR containing charges and tax)

When you enter rates into PortaBilling®100, you can define them in such a way that they incorporate the necessary charges and applicable taxes. For example, if your price is \$0.10 per minute and there is 20% VAT, the rate will be entered as \$0.12 (\$0.10 + 20%), as illustrated below. When the customer is charged, the total amount in the xDR will then include the appropriate amount of taxes. In our example below, the customer makes a 15-minute call, which is charged at \$1.80 – this amount includes both service charges and taxes.



When the invoice is created, the tax information must be properly presented to the customer. Since the total amount of the invoice and the tax rate are known, the actual amount of tax and the pre-tax amount can be “back-calculated”. In our example, we assume that there was only one charge during the billing period, so that the invoice total is also \$1.8; but it works exactly the same way if there are multiple transactions in the billing period, too. By applying the tax calculation formulas in the template, PortaBilling® determines that, since the invoice total is \$1.80 and the tax rate is 20%, the pre-tax amount was \$1.50 and the amount of tax \$0.30.

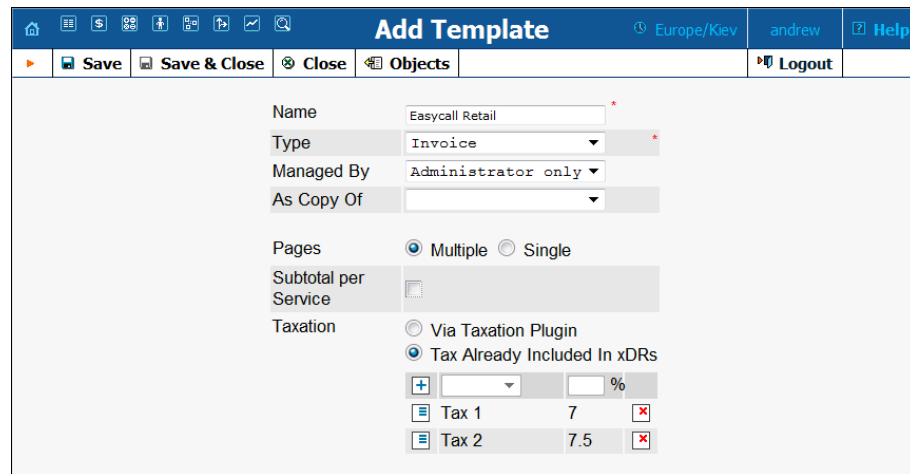
Rate	xDR	Invoice
\$0.12 (\$0.10 + 20%)	15 min	Services \$1.50
	\$1.80	Tax \$0.30
		Total \$1.80

This method is ideal for prepaid services. Since every xDR produced contains the tax amount, charges and taxes are debited from the

customer’s balance immediately after the service is rendered. This is also convenient for European countries, where customers are used to seeing all prices as “final”.

Invoice Templates

When creating a new invoice template, you will choose if this template is to be used to produce tax-inclusive or tax-exclusive invoices. If you choose the tax-inclusive option and specify the actual tax rates, the tax calculation formulas will automatically be entered in the post-processing rules in fields like Tax1, Tax2, and so on.



Name	EasyCall Retail
Type	Invoice
Managed By	Administrator only
As Copy Of	
Pages	<input checked="" type="radio"/> Multiple <input type="radio"/> Single
Subtotal per Service	
Taxation	<input type="radio"/> Via Taxation Plugin <input checked="" type="radio"/> Tax Already Included In xDRs
	+ [dropdown] [input] %
	Tax 1 7 %
	Tax 2 7.5 %

During invoice generation, PortaBilling® computes the tax amount based on the invoice total and the tax rate you have specified, and this information is displayed to the customer.

Back-calculation formulas support applying multiple types of taxes simultaneously. You simply need to enter the total tax rate and the rate of the specific tax. Our example above illustrates taxation for British Columbia (Canada), where there is 7% GST and 7.5% PST.

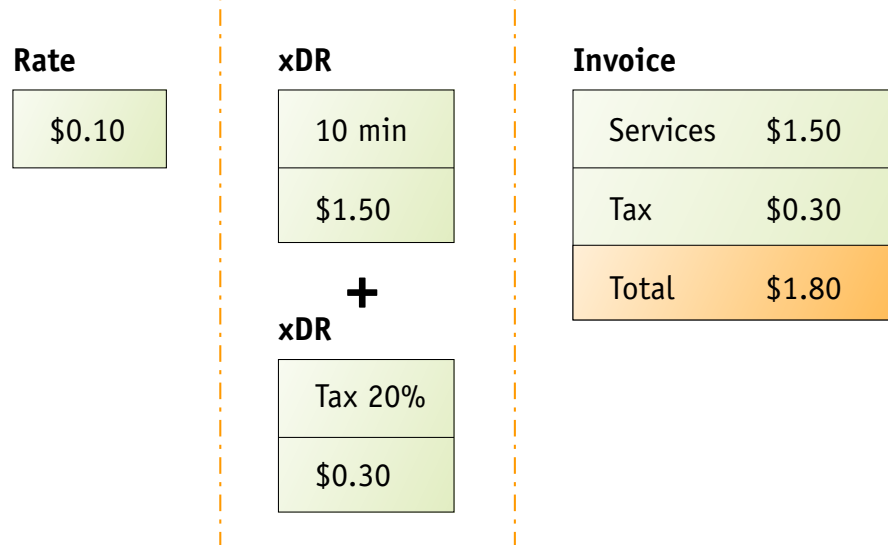
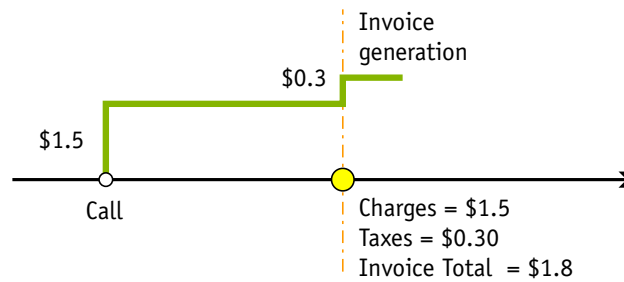
If you want to adjust the tax rate for an existing template or perform some other advanced modifications to the tax calculations, you can edit the contents of the post-processing rule formula directly. For more details about it and the **backcalcTax** function in particular, see the [PortaBilling® 100 Templates Guide](#) manual.

Exclusive Rate (separate xDRs for charges and taxes)

In this case, the prices you enter in the system exclude any tax information, and so when PortaBilling®100 processes a billing event, the charged amount recorded in the xDR is likewise exclusive of tax. Instead,

the taxation process is launched later, when closing the billing period for a given customer (this is actually the final step in closing the billing period process, after all other operations, such as charging recurring subscription fees, have been completed). All the customer’s transactions during the billing period are retrieved and analyzed, and the appropriate taxes are assessed. These tax charges are added to the customer’s account as separate records (xDRs), and then appear on the invoice along with the other charges. In this case, the invoice template does not perform any tax-related calculations; it simply presents the sum total of all “tax” and other transactions.

NOTE: Applying an invoice template designed for the inclusive method to a customer who is charged by the exclusive method will most certainly produce incorrect calculations.



This method allows greater flexibility when dealing with complex taxation (for instance, when the tax is not simply a percentage of the price). It is more suitable for postpaid customers, since tax amounts are applied to the balance only at the end of the billing period. It is also the preferred method for customers in the US and Canada, who are used to seeing prices without tax and having tax amounts added to a bill separately.

Since taxation rules differ from country to country, PortaBilling®100 supports a system of plug-in tax modules. Each customer is assigned his own plug-in type, which defines a different module to be used for calculating taxes. Currently, PortaBilling®100 supports plug-ins for VAT (fixed percentage) and the BillSoft EZTax suite. The latter enables correct calculation of US state, country and city taxes, as well as special items such as federal USF. For more details, visit the [BillSoft website](#).

Using the BillSoft EZTax Suite with PortaBilling

To use EZTax, you must first sign a contract with BillSoft (EZTax is a paid subscription service; please mention that you are a PortaOne customer to receive special pricing and other promotions) and obtain the EZTax libraries for installation on your server. After EZTax has been enabled in the PortaBilling®100 configuration, the following will be done for all customers with BillSoft assigned as the taxation method:

1. All accumulated transactions (xDRs, refunds, etc.) will be sent to EZTax along with the customer's information (which is used to determine his tax jurisdiction).
2. EZTax will calculate all applicable taxes and send them back to PortaBilling®100, so that they can be inserted as extra xDRs for the given customer (each type of tax will produce a separate record; thus if both state and city taxes are applicable, there will be two separate transactions).
3. PortaBilling® will then proceed to generate the invoice as usual.

In order to tax services properly, xDRs are mapped to EZTax's transaction/service codes, which define what type of taxes apply to a given transaction. The current logic of mapping XDRs is as follows:

1. If neither the CLI (origination number or ANI) nor the CLD (destination number or DNIS) record of the call event matches the North America Numbering Plan (NANP) format (i.e. the phone number should start with 1 followed by a 3-digit area code (NPA), then a 3-digit exchange prefix (NXX), and finally a 4-digit number), then this is considered to be an international call, and the **VoIP International Usage** (19, 51) transaction/service code is used.
2. If either the CLI or CLD matches the 1-NPA-NXX-NNNN pattern, but the given NPA-NXX combination is not found in the BillSoft address database (this usually means that, even though the phone number starts with 1, it belongs to some other country - for instance, the Dominican Republic), the **VoIP International Usage** (19, 51) transaction/service code is likewise used.
3. Then the location identifier for CLI and CLD are compared further:

- If the CLI and CLD are identified as belonging to different countries, the **VoIP International Usage** (19, 51) transaction/service code is applied (calls from/to Canada fall under this category).

NOTE: Calls between the USA and Puerto Rico are defined as calls of the interstate type, and the **VoIP Interstate Usage** (19, 49) transaction/service code applies.

- Then the **state** attributes in the location information are compared for CLI and CLD. If both CLI and CLD are reported as belonging to the same state, the **VoIP Intrastate Usage** (19, 50) transaction/service code is used; otherwise the **VoIP Interstate Usage** (19, 49) code is applied.

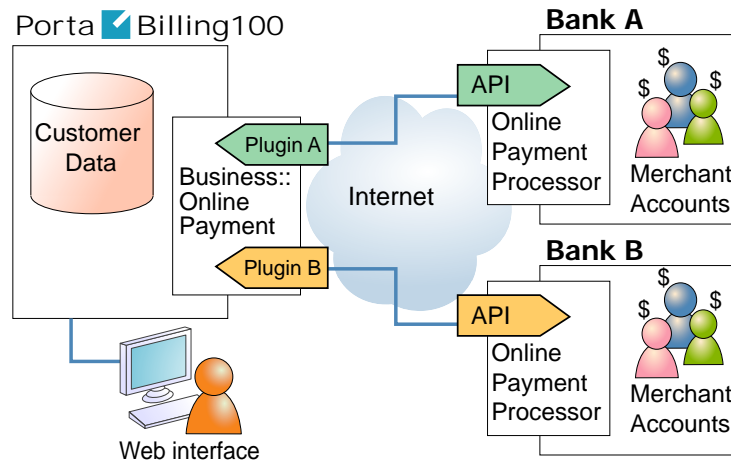
Payments

PortaBilling®100 allows payments to be processed online, without the intervention of an operator, by charging the customer's credit card or debiting his bank account. Payments may be initiated by:

1. Your subscribers from the web interface (called "online payments" because payment is triggered by the customer, who is using the web interface at the moment the payment is made).
2. The PortaBilling®100 system itself. These payments are called periodic, because they usually happen automatically over a regular period of time. There are three sub-types of periodic payments:
 - Calendar payments, which happen every month, every week, and so on. This is a convenient way to charge a customer his full balance, or some other pre-defined amount, at a certain moment in time.
 - Balance-driven payments, which are triggered by certain conditions involving a customer's balance. When the customer's balance is higher than the specified threshold, payment is initiated automatically.
 - Automated payment, which is applied when closing an invoice. The customer's credit card is charged for the invoice's amount due, so that when the invoice is created it shows an amount due of zero

Payment Flow

The following picture illustrates the individual components of the online payment system:



Merchant account

A merchant account must be opened at a bank. Its purpose is similar to that of a normal checking account. It stores funds you have received from credit card payments.

Online payment processor

A merchant account is usually enough to start processing credit card payments via charge slips. However, if you wish to initiate transactions from your own server via the Internet, this service is provided by online payment processors such as Authorize.Net. You establish an account with an online payment processor, providing it with your merchant account information, and in return you receive credentials (username, password, etc.) for using its API. So now your application can connect to the API's server and, upon providing valid authentication information, initiate transactions.

Business::OnlinePayment framework

Unfortunately, there are many different online payment processors, most of them using their own API which is different from that of other providers. So it seems that every e-commerce application such as PortaBilling@100 must be programmed to support them all, which is an extremely laborious task. Fortunately, there is a solution: Business::OnlinePayment is a framework which provides the necessary middleware between e-commerce applications and online payment processors.

- The application is provided with unified API methods, e.g. doTransaction, so the same code in the application can be used to communicate with any online payment processor.
- A communication plugin is created for each online payment processor. This plugin knows such specific parameters as the server's address, the correct way of encoding transaction

parameters, the proper way of parsing the server's response, and so on.

On the application side, therefore, only a few things are required: loading the correct plugin and invoking a transaction method. After that control is handed over to the plugin, which handles all communication with the remote server.

The whole Business::OnlinePayment project is open-source, and is maintained by a large team of individuals. At the moment about 10 different plugin modules are supplied with PortaBilling®, and it is easy to create new plugins for online payment processors. You can find more information about the Business::OnlinePayment module and available plugins at <http://www.cpan.org/modules/by-module/Business/>.

PortaBilling® payment support

PortaBilling® stores required information, such as customer name and address, credit card number, and so on, in the database. Multiple merchant accounts are supported (for instance, your merchant account A accepts payments in USD, while account B takes payments in euros). You may define a “minimum allowed payment” and a list of supported card types (VISA, MC, etc.) for every merchant account.

When payment is initiated (either by the customer from web self-care, or automatically) the system connects to the online payment processor and performs the transaction. The online payment processor verifies the credit card information (and performs optional steps such as address verification or CVV control). Upon receiving confirmation that the transaction was successful, PortaBilling® writes transaction information to the PortaBilling® database and modifies the balance for the account/customer.

Supported payment methods

The choice of available payment methods depends on the online payment processor. In general, however, PortaBilling®100 supports the following payment methods:

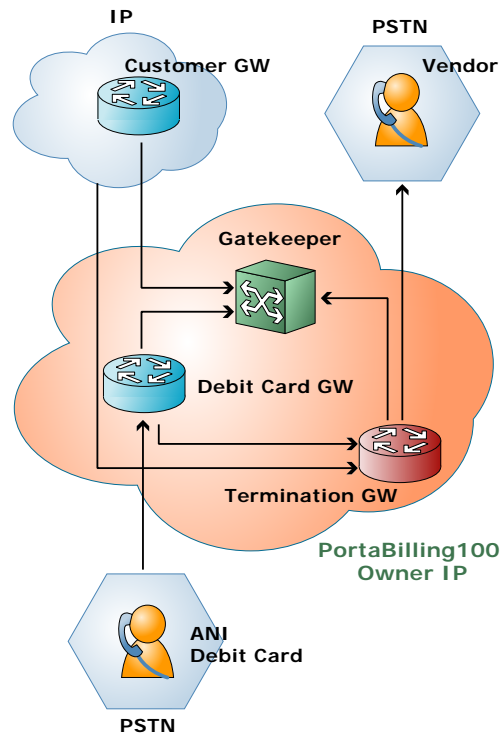
- MasterCard
- VISA
- American Express
- Discover
- FirePay
- Switch/Maestro
- E-check (direct debit of bank account)

4 . Voice Services

CDR Mediation

PortaBilling® can collect information about the call via various network elements (e.g. PortaSIP and the media gateway), combine the individual pieces and produce a single CDR record.

Collecting data from the VoIP network



Gateways in the PortaBilling®100 owner network are “trusted” gateways. These gateways **MUST** be listed in the Nodes directory and be a RADIUS client.

Gateways which belong to your customers or vendors should not be created as nodes. Instead, a remote GW which belongs to your customer will be represented in the system as an account.

If the termination GW transmits an accounting record (connection matched), the User-Name must be verified to see if a trusted GW (Node IP=User-Name) is calling. If so, then the billing engine must wait for an accounting record from the trusted GW and use the User-Name from a trusted GW accounting record. If a trusted GW is not found, the call will be charged immediately to an applicable account. If an account is not found, a “security breach” mail alert will be sent.

Call Legs

Billing for telephony services is done based on the information provided by the elements of your VoIP network: gateways, SIP server, and so on. This information is supplied to the billing engine as accounting records. Usually the gateway treats incoming and outgoing calls as two separate entities, i.e. **call legs**. Each of the call legs produces a separate accounting record, so that, for instance, billing engine receives 4 accounting requests in case of a “classic” VoIP call arriving from PSTN to gateway A, traveling over the Internet to gateway B, and terminating in the PSTN network once again. Of course there must be some way of knowing that these four records are actually part of the same logical call. Therefore, each of them includes a special attribute that carries a unique ID for the call. This ID is used by the billing engine to add the call legs together, and is also extremely useful for troubleshooting. For H323 calls, this attribute is called `h323-conf-id`, while for SIP calls it is `call-id` (however, for backward compatibility Cisco gateways and PortaSwitch® assign a `h323-conf-id` for every SIP call).

Call Routing

When an entity on your network (for instance, a gateway or a SIP server) has to establish an outgoing call, it must find out where the call is being sent to. To do this, it can use its internal configuration (for example, dial-peer on the gateway) or some external entity (e.g. 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).

Currently, the following VoIP nodes can obtain routing information directly from billing:

- PortaSIP® server (for SIP calls)
- MVTSPPro (for H323/SIP calls)

Control of routing on the Cisco gatekeeper (via GKTMP) or other equipment is not currently supported, but you can develop your own routing interface for PortaBilling®, which will communicate with PortaBilling® using RADIUS and then transfer this information to the originator in the correct format.

High-level Routing Algorithm

The following happens after the customer making the call has already been identified, the phone number has been translated into E.164, and all the required authorization checks have been performed:

- If the call is for one of the phone numbers provisioned in the system (there is an account with an account ID identical to the dialed number), direct the call to the PortaSIP® server where the SIP phone is currently registered. Also, provide a list of routes for follow-me numbers or a voice mailbox, so that the call can be redirected there if the SIP phone is not currently available (off-line, not answering, etc.).
- If the call is made to one of the access numbers for IVR applications, route the call to the node (PortaUM®, PortaBridge, etc.) which runs that application.
- Otherwise, route the call to one of the gateways for termination, according to the routing rules specified in PortaBilling®.

Routing of SIP On-net (SIP-to-SIP) Calls

As the first step in routing calculations, PortaBilling checks if there is an account with an account ID equal to the dialed number (given in E.164). If such an account is found, it is clear that the call is designated for one of the subscribers provisioned on the system. PortaBilling then extracts the latest known location for this account from the location database, in order to determine which PortaSIP® server the account is currently registered on, and routes the call there. The SIP server maintains information about all currently registered SIP user agents, so it is able to properly communicate with the SIP user agent handling this number.

Routing of Access Numbers

This is a special case of “internal” routing, when a call does not leave the network. When a customer dials (from a SIP phone or the external PSTN network) an access number assigned to an IVR application on PortaUM® or PortaBridge, the call has to be routed properly. PortaBilling maintains a table of such access numbers, indicating where they should be routed to.

Routing of Off-net Calls

You can have different vendors for terminating 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 with the **Routing** type, a few more fields will be added to rates in that tariff:

- **Route category** – you can split this into categories such as “Premium”, “Cheap”, etc. and use these categories in routing plans

- **Preference** – routing priority (0-10), higher values mean higher priority, 0 means do not use this rate for routing at all
- **Huntstop** – signals 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 it 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 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 the 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 Premium and then Normal route categories
- **Ordinary** - includes first Normal and then Premium route categories
- **Cost-efficient** – includes only Normal route category

Thus the system will offer a different set of routes depending on which routing plan is assigned to the account making a phone call. One routing plan, called **All Available Routes**, is always in the system, and is assigned to the account unless otherwise specified. This routing plan uses all available routes, regardless of their route category.

Routing algorithm

The routing principle is simple:

- The SIP server (or MVTIS, or some other entity) asks PortaBilling® for a list of routes for a given number.
- PortaBilling® checks every tariff with routing extensions associated with a vendor connection to find rates matching this phone number. The best-matching rate is chosen in each tariff; this rate defines the routing parameters, i.e. it supplies a list of all vendors who can potentially terminate this call, including their cost and other related parameters.
- A list of possible termination addresses will be produced (this will include remote IP addresses for VoIP connections and IP addresses of your own nodes with telephony connections).
- This list will be sorted according to route category, routing preference and cost. Entries assigned a route category not included in the routing plan and entries which do not satisfy the profit-guarantee criteria will be removed; entries with a preference lower than the one in the first huntstop entry will be ignored.
- If adaptive routing is used, then routes to vendors currently penalized for not meeting the quality criteria will be moved to the very last positions in the routing list.
- If there are several routes with an identical cost/preference, load sharing will be applied so that each potential route has an equal chance of being the first. Consequently, if your termination carrier provides you with three gateways to send calls to, each gateway will, in the final analysis, receive approximately the same number of calls (33%).
- A list of these IP addresses (with optional login and password for SIP authentication) will be returned to the SIP server. (To avoid extremely long call setup time, only a certain number of routes from the beginning of the list are returned; the default is 15, but this can be changed in the configuration).

Route sorting

How exactly does PortaBilling®100 arrange multiple available routes?

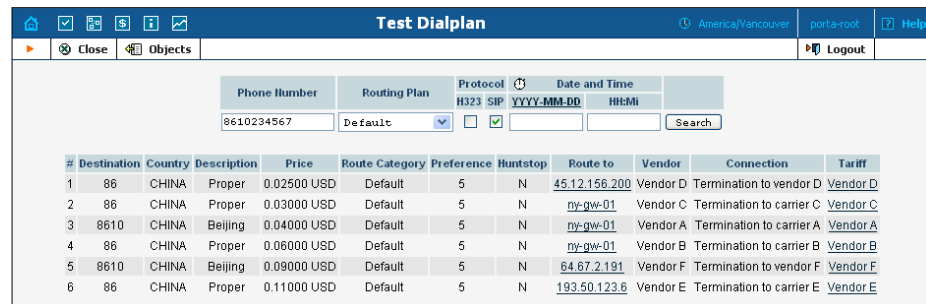
1. By route category. Only route categories which are included in the routing plan will be used, following the order given in the routing plan.
2. If you have multiple route categories within the routing plan, you can either merge them into the same group by assigning them the same order value, or keep 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. For routes with the same preference, the system can arrange them according to cost (a comparison is made on the **Price_Next** rate

parameter) so that cheaper routes will be among the first ones, or in random fashion.

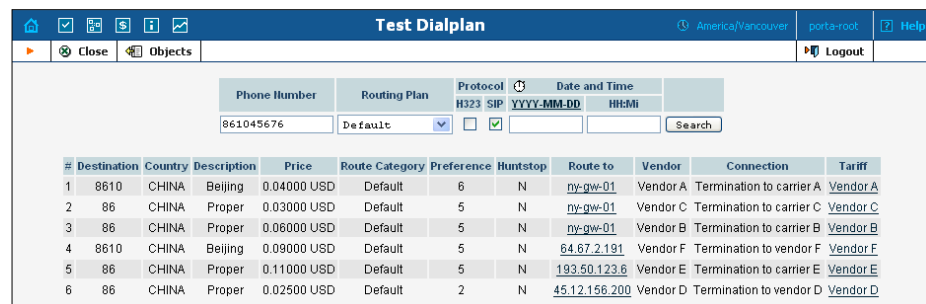
Does PortaSwitch support LCR?

Yes, PortaSwitch® supports LCR – and much more besides. In fact, “just LCR” is the simplest type of routing PortaSwitch® 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 affected solely by the vendor’s cost. Also, PortaBilling®100 supports a “profit-guarantee” mode for routing. Only those termination carriers who satisfy your conditions for the minimum required profit (e.g. at least \$0.005 each minute) will be chosen.

Compare the two illustrations below. The first one shows simple least-cost routing: all of the available carriers are arranged according to cost. In the second illustration, PortaSwitch® routing preferences are used, so that the administrator can re-arrange the routing. In this case, carrier D has been moved down the routing list, while carrier A has been moved up to the top of the list, thus becoming the first route.

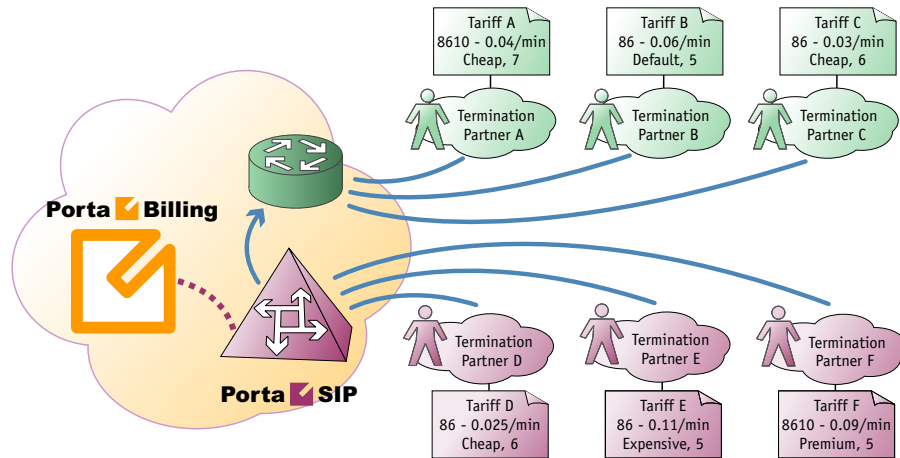


#	Destination	Country	Description	Price	Route Category	Preference	Huntstop	Route to	Vendor	Connection	Tariff
1	86	CHINA	Proper	0.02500 USD	Default	5	N	45.12.156.200	Vendor D	Termination to vendor D	Vendor D
2	86	CHINA	Proper	0.03000 USD	Default	5	N	ny-gw-01	Vendor C	Termination to carrier C	Vendor C
3	8610	CHINA	Beijing	0.04000 USD	Default	5	N	ny-gw-01	Vendor A	Termination to carrier A	Vendor A
4	86	CHINA	Proper	0.06000 USD	Default	5	N	ny-gw-01	Vendor B	Termination to carrier B	Vendor B
5	8610	CHINA	Beijing	0.09000 USD	Default	5	N	84.67.2.191	Vendor F	Termination to vendor F	Vendor F
6	86	CHINA	Proper	0.11000 USD	Default	5	N	193.50.123.6	Vendor E	Termination to carrier E	Vendor E



#	Destination	Country	Description	Price	Route Category	Preference	Huntstop	Route to	Vendor	Connection	Tariff
1	8610	CHINA	Beijing	0.04000 USD	Default	6	N	ny-gw-01	Vendor A	Termination to carrier A	Vendor A
2	86	CHINA	Proper	0.03000 USD	Default	5	N	ny-gw-01	Vendor C	Termination to carrier C	Vendor C
3	86	CHINA	Proper	0.06000 USD	Default	5	N	ny-gw-01	Vendor B	Termination to carrier B	Vendor B
4	8610	CHINA	Beijing	0.09000 USD	Default	5	N	84.67.2.191	Vendor F	Termination to vendor F	Vendor F
5	86	CHINA	Proper	0.11000 USD	Default	5	N	193.50.123.6	Vendor E	Termination to carrier E	Vendor E
6	86	CHINA	Proper	0.02500 USD	Default	2	N	45.12.156.200	Vendor D	Termination to vendor D	Vendor D

Routing configuration example



Consider the following example. If you have:

1. A “Standard” routing plan, which includes three 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 routing 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).

Test Dialplan											
Phone Number		Routing Plan		Protocol		Date and Time					
8610234567		Standard		H323 SIP		YYYY-MM-DD		HH:MM		Search	
#	Destination	Country	Description	Price	Route Category	Preference	Huntstop	Route to	Vendor	Connection	Tariff
1	86	CHINA	Proper	0.06000 USD	Default	5	N	ny-gw-01	Vendor B	Termination to carrier B	Vendor B
2	8610	CHINA	Beijing	0.04000 USD	Cheap	7	N	ny-gw-01	Vendor A	Termination to carrier A	Vendor A
3	86	CHINA	Proper	0.02500 USD	Cheap	6	N	45.12.156.200	Vendor D	Termination to vendor D	Vendor D
4	86	CHINA	Proper	0.03000 USD	Cheap	6	N	ny-gw-01	Vendor C	Termination to carrier C	Vendor C
5	86	CHINA	Proper	0.11000 USD	Expensive	5	N	193.50.123.6	Vendor E	Termination to carrier E	Vendor E

Load Balancing

The routing list, computed according to the costs and preferences of individual vendors, is “static” – meaning that each vendor will always maintain his respective position in the list (first route, second route, last route, etc.) *unless* the vendor changes his cost or the administrator manually adjusts the preference for that vendor. Let’s look at the following example:

Vendor	Preference	Cost (\$/minute)
A	5	0.05
B	5	0.06
C	5	0.15

In this case, after the sorting is done according to cost, Vendor A will occupy the first position in the list – and the first routing attempt will always be made to that vendor. Only if the call cannot be connected to Vendor A will it then be routed to Vendor B. This basically means that under normal circumstances, Vendor A will get most of the traffic, and a call will be routed to Vendor B only if all available capacity of Vendor A has been exceeded or if there is some temporary problem with connecting the call. Although this scheme offers maximum profit, conducting business this way is usually undesirable in real life. If the cost of several routes is comparable, it is better to allow a portion of the traffic go further down the routing list (via other carriers; Vendor B in our example) to ensure that the other routes are maintained in an active state, that there is a sufficient amount of data for producing quality reports and that our contracts with those vendors will not be canceled due to inactivity.

The idea is to allow a portion of all traffic to be directed to the vendors who are further down the routing list – though of course, this can only be done if the cost difference between the vendors is acceptable. In our example, Vendor C has a price per minute of \$0.15 – so this route should obviously not be mixed with those of Vendors A and B, since the difference in cost (\$0.10/min) is unacceptable.

By setting up the **load balancing threshold** in the routing plan, the administrator specifies the maximum acceptable difference in cost between two carriers, thereby indicating when routes to these carriers can potentially start “swapping” positions in the routing list.

How does this value affect the route sorting in exact terms? After the list of potential routes is produced, but before the actual route re-arrangement according to cost takes place, each route cost is slightly changed by a random value. The adjusted cost may be increased or decreased compared to the original one, but the spread between the minimum possible value and the maximum possible value will be specified as the **load balancing threshold**. For instance, when the load balancing threshold is set to \$0.02, then Vendor A’s cost (\$0.05) may be increased or decreased by \$0.01 (one half of \$0.02), changing the cost to a random value between \$0.04 and \$0.06. Naturally this adjusted cost is purely fictional; these changes would only take effect during the sorting of routes and would not affect the actual billing of these calls.

As a result of these cost fluctuations, when the list of routes is sorted according to cost, the route sequence “shuffles” and the probability of two routes swapping places increases when they have similar prices. If in our example, above, we set the load balancing threshold to \$0.02, the adjusted cost of A would be in the range of \$0.04 and \$0.06, and the adjusted cost for B would be in the range of \$0.05 and \$0.07. If for instance, the new cost of A happens to be \$0.053 and the adjusted cost for B happens to be \$0.052 –Vendor B would move into the first route position.

Overall, in about 25% of all cases both values would be randomly distributed in the same range (\$0.05-\$0.06) and therefore, B would become the first route for about 12,5% ($25\% / 2$) of all call attempts. In comparison, if B’s cost were \$0.051 – this algorithm would result in B receiving an almost equal share (49.5%) of all calls that A is receiving, since adjusted cost values for A and B would be randomly distributed throughout two nearly 100% overlapping intervals (\$0.04-\$0.06 and \$0.041-\$0.061) and thus each vendor would have nearly the same chance of his adjusted cost being the lowest.

This process of randomly adjusting the cost followed by re-arrangement is done for every new call. So for each call the actual routing list may look slightly different from previous lists and the overall call volume is properly distributed among multiple vendors.

Note that load-balancing only applies to routes with the same preference and route category that have been assigned the same order in the routing plan. This is to prevent load-balancing (when undesirable) by simply increasing or decreasing the preference of a specific route. So if in our example the route to Vendor A has a preference of 6 and the route to



Vendor B still has a preference of 5 – load-balancing would not happen, and A would remain in the first route position.

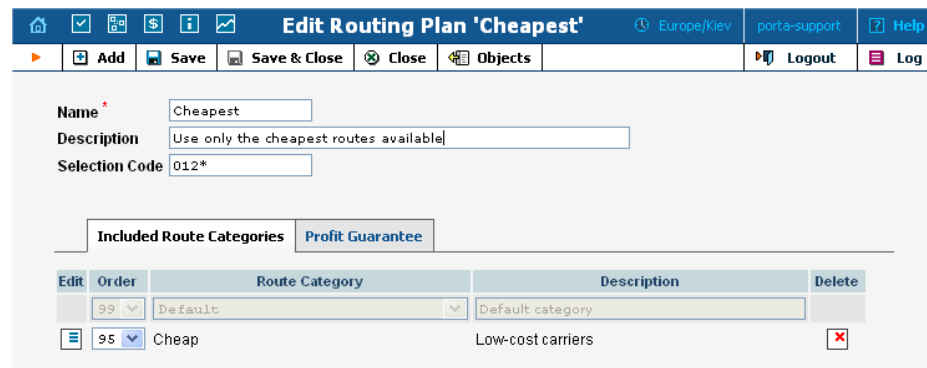
Routing Plan Selection

There are two possibilities for choosing which routing plan is used to find available vendors and terminate a call:

- Each account is assigned a default routing plan. When the end-user simply dials a number in the normal way, this routing plan will be used.
- If the end-user wants to choose a specific routing plan for a particular call, he dials a routing plan selection code before the number. For example, if he wants to use the “Premium” routing plan to call US phone number 12065551234, he would dial 23*12065551234, where 23* is the selection code defined as a property of the “Premium” routing plan.

Selection codes



The selection code is a property of a routing plan, and so obviously each routing plan has its own unique selection code. If you want to have a routing plan which cannot be specifically selected by an end-user, simply leave the selection code field empty.



The screenshot shows the 'Edit Routing Plan' interface for a plan named 'Cheapest'. The interface includes a header with navigation icons and a toolbar with buttons for Add, Save, Save & Close, Close, Objects, Logout, and Log. The main form contains the following fields:

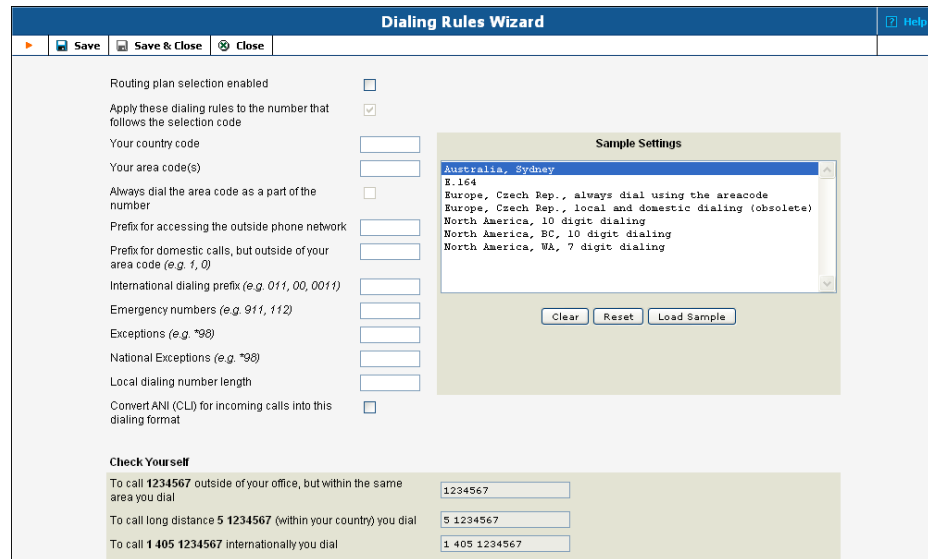
- Name:** Cheapest
- Description:** Use only the cheapest routes available
- Selection Code:** 012*

Below these fields are two tabs: 'Included Route Categories' (selected) and 'Profit Guarantee'. Under the 'Included Route Categories' tab, there is a table with the following data:

Edit	Order	Route Category	Description	Delete
	99	Default	Default category	
	95	Cheap	Low-cost carriers	

Dialing Rules

By including a routing plan in the accessibility definition, you potentially allow every account with this product assigned to it to use this routing plan. If you want to prohibit a specific customer from selecting this routing plan for a call, you can disable it in the customer’s dialing rules. In that case, calls will be routed using the account’s default routing plan.



Required support outside PortaBilling

In order to be able to use dynamic routing plan selection and have these calls charged properly, adequate support is required on the switch side, i.e. the node which processes these calls. This derives from the fact that, although the customer initially dials the selection code, this information is not included in the phone number sent to the vendor, and therefore is not visible in the call accounting information. Therefore, the node must provide this information to PortaBilling as a separate attribute, so that the call can be charged properly.

PortaSIP® supports this functionality by default, while for switching equipment supplied by a third-party vendor it is recommended to consult with the vendor directly.

Adaptive Routing

PortaBilling® allows you to automate the process for controlling the call quality which becomes increasingly important today. That is, if you want to evaluate acceptable vendors for terminating VoIP calls, there is no need to hire numerous human operators or network engineers, who will track and analyze the specific route. All you need is to implement adaptive routing model.

The main idea of adaptive routing is to dynamically measure a vendor’s quality parameters, and adjust the routing priority accordingly. These quality requirements are predefined in the form of threshold parameters on the **Routing Criteria** page, and are then automatically applied to specific vendors. Any vendor who fails to satisfy your quality requirements will go to the “penalty box” – the very bottom of the routing list. This means that the system will try first to terminate calls using other carriers (with a good quality evaluation). However, if all of





them fail or are unavailable, the “penalized” carrier will have a chance to terminate the call. It is still better to send a call via an inferior vendor than to have it fail completely.

Thus, if the quality requirements are applied, a carrier’s place on the routing list is determined not only by the route category, the assigned preference value, and the cost parameters (LCR model), but additionally by quality criteria.


For effective quality measurement, the following key control parameters are used:

- ASR (Average Successful Rate) – The number of successfully connected calls divided by the total number of call attempts.
- PDD (Post Dial Delay) – The time interval between the connection request to a vendor and ring-back.
- ALOC (Average Length of Call).
- PPM (Profit per Minute) – The aggregated profit, i.e. the difference between the actual charged amounts in the CDRs for your customers and vendors.

Each of the above parameters requires two values, which define the warning and penalty thresholds, respectively. When the value of a parameter reaches the predetermined threshold, the administrator receives an e-mail alert about the latest connection threats. Moreover, the administrator can track the current connection status on the **Tracking** page. This status is represented by different colors, as follows:

-  GREY – the number of calls is not enough to apply filtering differentiation;
-  GREEN – the route meets the quality requirements;
-  YELLOW – the route is active, but some of its quality parameters are outside the warning thresholds;
-  BLOCKED - this route is currently being penalized.

NOTE: The penalized route will stay in the “penalty box” for the period of time specified in **Penalty Time**, and then will be unblocked automatically. Alternatively, you can unblock the penalized route manually by clicking the **Unblock Now** button.

-  RED – the route was manually unblocked; this status will remain until the next time interval for computing statistics.

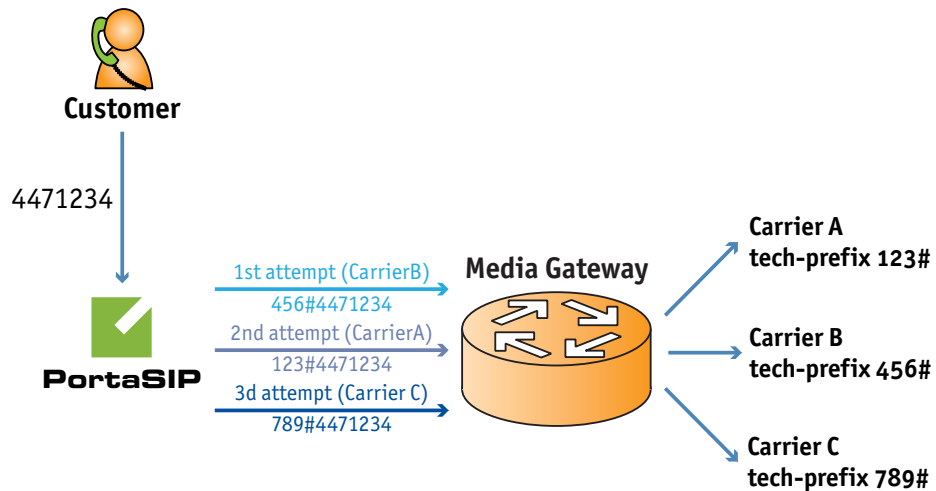
Routing to an external media gateway

A common situation is when several different carriers are connected via PRI trunks (or SS7 interface) to the same media gateway, which does the conversion from VoIP to PSTN. When a call is sent to the gateway, the

question is which particular carrier should be used to connect this particular phone call (or if multiple carriers are used – in which order should they be tried)?

Most of the gateways support configuration of routing for outgoing calls via dial-peers or similar method, where the outgoing carrier or trunk group is chosen based on matching the dialed number (e.g. 4471555123456) to a set of prefixes (e.g. 447 or 4471) assigned to outgoing trunks. So it is possible to maintain the routing configuration on the gateway, but it is not taking into consideration the costs of sending calls to a particular vendor, and any change in the routing requires manual re-programming of the routing table on the gateway. Needless to say this method significantly increases the amount of work for the administrator, reduces routing flexibility and is very likely to generate routing errors, which could seriously damage revenue of the service provider, since calls are not being sent to the carrier with lowest costs.

So another approach is recommended: the media gateway is configured with the bare minimum of the routing information. Each outgoing carrier (or trunk group) is assigned its own prefix (e.g. 123# for carrier A and 456# for carrier B). When PortaSIP sends a call to media gateway, it appends the specific carrier’s prefix to the dialed number, so only that carrier is used to transport this call (as illustrated below).



As a result the call routing is controlled at the single location, and one has to maintain a single set of carrier rates and related configuration

Routing in a multi-node scenario

Sometimes a network consists of multiple nodes (for instance, if you have two PortaSIP servers and one server with MVTSP installed), each of them capable of doing call routing under the control of PortaBilling. In this case, two scenarios are possible:

- The node which receives the original call from the customer proceeds with routing of the call and delivers it to the “destination” carrier. This is the simplest scenario, and usually it provides the best results in terms of your network’s usage (since the call always travels the shortest possible path, with a minimum number of hops on the network). In this case, though, each node communicates with the carrier directly, which may require additional maintenance activities on your side (e.g. if the carrier uses authorization by IP address, you will need to provide him with a list of all your nodes’ IP addresses and keep that list updated as your network changes).
- The node which receives the original call from the customer does the authorization (to verify the identity of the caller) and sends the call over to the “master” node. The master node then obtains the actual list of routes for this call and establishes the connection to the terminating carrier.

The last scenario increases traffic within your network, but it allows you to handle advanced scenarios – for instance, when you have multiple PortaSIP nodes deployed in different countries (and their IP addresses change often), but only your central PortaSIP node, located in your main collocation facility, is able to communicate with the carrier because of IP address restrictions. Another case where this scenario may be useful is when the majority of your carriers are H323. In this case, it would make sense to send calls from the individual PortaSIP nodes to MVTSP (master node), which will then route the calls onward.

Routing to carriers, which only support H323

If the number of H323-only carriers is not too high, it would make sense to configure the system as follows:

You can use Cisco IPIPGW, MVTSP, or any of the SBCs available on the market as a SIP-H323 convertor. The idea is to configure it in such a way that any incoming SIP call where the destination number starts with a certain tech-prefix will be sent out to a particular carrier. For example, if the destination number in the incoming call starts with 1#, the call will go out to IP address 6.7.8.9 (SuperNet); if the destination number starts with 2#, to IP address 1.2.3.4 (X-Telecom); and so on. Although this converter

is (strictly speaking) part of your network, there is no need in this case to define it as a node in PortaBilling, since it will not be exchanging any information via RADIUS.

In PortaSwitch you will configure your SIP carriers as usual. When you create a connection for an H323 carrier, you will specify the IP address of your SIP-H323 converter as the **Remote IP Address** and then assign a tech-prefix to this connection. In this way, although calls for multiple carriers will go from PortaSwitch to the same IP address, the system can still recognize which carrier is used in each case.

Let's consider the following example for a better illustration of how this process works. Assume you have 4 carriers, all capable of terminating calls to the UK. Their respective rates per minute are: carrier A - 0.02, carrier B - 0.03, carrier C - 0.05 and carrier D - 0.10. Carriers B and D are H323 only, so for them connections are configured on the SIP-H323 converter (with IP address 192.168.0.3), with tech-prefix 2# and 4#, respectively. A customer dials phone number 4412345678. The call arrives to the PortaSIP node, which, upon successful authorization, receives the following routing list:

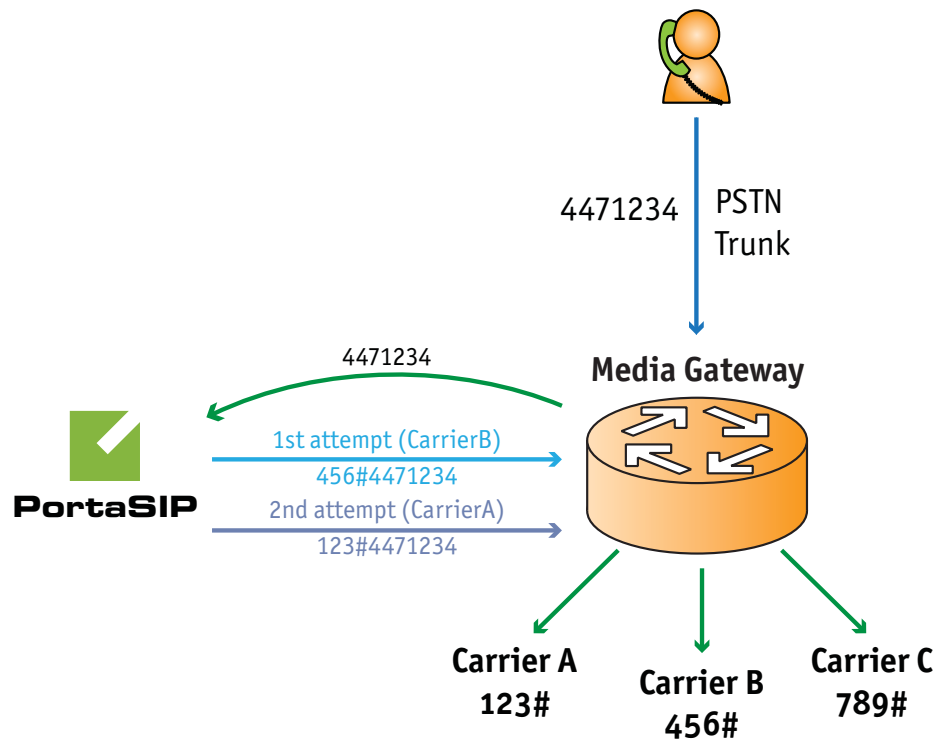
```
Route 1: 4412345678 @ <IP address of carrier A>
Route 2: 2#4412345678 @ 192.168.0.3
Route 1: 4412345678 @ <IP address of carrier C>
Route 1: 4#4412345678 @ 192.168.0.3
```

PortaSIP will first try to establish a connection via the first route and, if that fails, it will send a call via the second route, to the SIP-H323 converter, which in turn will attempt to establish a call with carrier B. If a connection is not possible, after receiving an error message from the converter PortaSIP will try the next available route, and so on.

This method of configuration allows the SIP-H323 converter resource to be used only when it is actually required (since SBCs are usually licensed on a per-port basis), thus minimizing total network infrastructure costs.

Intelligent routing on a third-party media gateway

A common situation faced by service providers today is when a piece of communications equipment on the network (VoIP/TDM switch, media gateway, SBC, etc. – referred to below as **the switch**) is not capable of doing real-time call authorization or intelligent call routing like PortaSwitch. How can you still perform call processing on such equipment in conjunction with the advanced call control and routing options that PortaSwitch offers?



In this scenario, the most straightforward method would be the following:

- The switch is configured to accept an incoming call initiation request and send it to PortaSIP.
- PortaSIP receives an incoming call and sends a request to PortaBilling to perform caller authorization and obtain the routing list.
- If authorization fails, the call disconnect command is sent back to the switch, and the incoming call on the switch is dropped.
- In case of successful authorization, PortaBilling computes the routing list (using LCR, preferences, settings in the customer’s individual routing plan, the current status of adaptive routing information for carriers, etc.). The resulting list of routes is delivered to PortaSIP, which then starts processing them. If there is a carrier directly connected to the switch, PortaSIP sends the call back to the switch, adding a tech-prefix to the destination number to indicate the specific trunk group this call should be routed to. If the first route fails, another call is sent to the switch (indicating a different trunk group), and so on. When the call is established, PortaSIP acts as a signaling-only node, so there is no effect on the codecs, sound quality, or other voice media attributes. Finally, when the maximum allowed time has been reached, PortaSIP sends a call disconnect request to the switch.

This allows the use of call authorization, advanced real-time routing and control of the customer’s credit limit, even with equipment which does not directly support these options.

Number Translation

Very often you need to use different numbering systems on your VoIP network. For example, Vendor A requires you to send all phone numbers with the technical prefix **12345#**, Vendor B requires that all phone numbers be in the format **011<country-code><area-code><number>**, and when you send calls to the local telecom (Vendor C) you have to strip the country code and area code off completely. Of course, you would like to have this information in a unified format in your billing, so that you do not need to enter multiple phone prefixes for the same destination. Another common situation is when customers located in different countries or regions (but all managed on the same PortaBilling® platform) dial phone numbers in different formats, since every customer dials a number in the way he or she is used to.

Thus the solution is to maintain all the numbers inside your network and PortaBilling® in a unified format – E.164 – and use **translation rules** to deal with situations when numbering formats differ. When a customer makes a phone call, it arrives from outside your network and the translation rule specific to that customer is applied. Thus the dialed number is converted from the customer's format (e.g. 0114202123456) into the unified format (4202123456). Next, all the billing calculations (authorization, routing, etc.) are done, and then, if it appears that this call should go to a vendor who requires a special format, the outgoing translation rule associated with this vendor's connection is applied, and the number is translated into the required format (e.g. 123#4202123456). Finally, when the call is disconnected and PortaBilling® receives information about the completed call to phone number 123#4202123456, the vendor's translation rule is applied, so that the number becomes 4202123456. This is the number that will be stored in the xDR information.

A translation rule is actually a set of Perl regular expressions which are applied to a phone number. So if you are familiar with regular expressions, you can make all kinds of transformations with phone numbers; for more information, see the PortaBilling® WebReference Guide. If you are not familiar with regular expressions, you can use a dialing rule wizard which will construct the correct rule based on the parameters you provide, such as country or area code. This allows you to easily manage a network with many different numbering plans for your customers and vendors. Also, translation numbers can be applied to ANI (CLI) numbers, allowing caller information to be sent to a vendor in the format he requires (e.g. 10-digit phone numbers for US callers).

Your network numbering plan

The key to avoiding problems with number formats is to choose a certain number format as the standard for your network and make sure that calls

travel on your network only in this format. The ideal candidate for such a format is E.164 (of course it is highly recommended that you use this same format in billing as well).

Customer-based translation rules

Frequently a customer will have his own numbering format, for example, dialing 00 for international numbers and just the phone number for local calls. Customer-based translation rules allow you to convert a number from a format specific to a particular customer. Customers can even manage their dialing format themselves via the self-care interface.

Customer-based translation rules have two applications:

- When a number is submitted for authorization, these rules will be applied, with the resulting number used to search rates. Thus, if your customer dials 0042021234567, you can convert it to 42021234567 and find the correct rates for the 420 prefix.
- This number will be returned to the node which requested it.

Connection-based outgoing translation rules

If your vendor requires a special number format (e.g. tech-prefix), it is possible to apply this rule to convert the number. When billing returns a list of routes, the phone numbers for routes for this connection will be converted. This only works for a routing model in which the VoIP node (e.g. PortaSIP®) requests billing for routing information. If your gateway uses dial-peers or an external gatekeeper for routing, then you must configure number translation there.

Connection-based translation rules

When the call has been terminated to the vendor in a vendor-specific format, it will be reported to billing in this same format (e.g. 7834#42021234567). Now it is necessary to convert this number to the proper format for billing (4202134567), which may be done using connection translation rules. These rules will be applied to all calls which go through a given connection (even those routed there using dial-peers or other external tools).

Connection translation rules are also used on **VoIP from Vendor** connections, when a number received from a vendor (e.g. 2065551234) must be translated into E.164 format, so that it will properly match an account whose ID is in E.164 format (12065551234).

Node-based translation rules

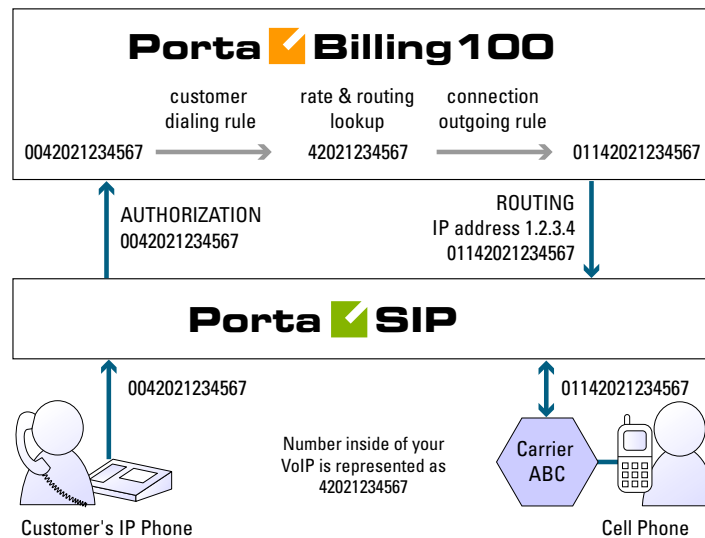
These serve the purpose of converting a number from a custom format used by the customer into billing's internal format during authorization, depending on the gateway. For example, on a gateway in Prague, Czech Republic, there may be the translation rule "strip leading 00", while on a

gateway in Moscow, Russia, the rule will be “strip leading 810 or replace leading 8 with 7”.

Currently, node-based translation rules are only kept for backward compatibility with older releases, since they have been made obsolete by customer translation rules. They will most likely be removed from future releases.

Number translation on your network

Below is an illustration of how different translation rules are applied during a call.



1. The customer dials a phone number on his SIP phone. He enters the number in the same format he uses on his conventional phone, i.e. 0042021234567.
2. The number is delivered to the PortaSIP® server and translated using the **customer’s dialing rule**, which states that the international dialing prefix for this customer is 00. So the number becomes 42021234567 (E.164 format). This number is used to search for the customer’s rate for this destination.
3. PortaSIP® then requests routing for this number. Carrier ABC is defined for terminating calls to the Czech Republic in PortaBilling®. However, this carrier requires the number to be in US dialing format, so the international number must be prefixed by 011. An **outgoing translation rule** `s/^/011/;` to carrier ABC has been defined, and is now applied to the phone number, with the result 01142021234567. Note that there may be several carriers who can terminate this call, each with its own numbering format. In such a case there will be several alternative routes with different phone numbers.

4. PortaSIP® attempts to establish a connection to remote gateway 1.2.3.4 using phone number 01142021234567.
5. After the call is completed, PortaSIP® sends an accounting request to PortaBilling®, stating that a call to remote gateway 1.2.3.4 has just been completed. PortaBilling® finds a connection to vendor ABC with remote IP address 1.2.3.4, and applies the **translation rule** `s/^011//`; for this connection in order to convert the number from the vendor-specific format into your billing format. Thus 011 is removed from 01142021234567, and the number becomes 42021234567. PortaBilling® searches for the vendor and customer rates for this number and produces the CDRs.

Local Number Portability and Other Special Cases

While call processing (routing, billing) based on a phone number match to a phone prefix (when each phone prefix defines a group of phone numbers such as 420 – Czech Republic; 4202 – Czech Republic, Prague; 420604 – Czech Republic, T-Mobile and so on) is used in the majority of telecommunications services, in some cases it is not applicable.

Local Number Portability

This is a very common example of a situation where the rate or call routing cannot be determined simply by looking at the phone number itself. Local number portability (LNP) is widespread throughout Europe, the US and other countries around the world. It means that when a user migrates from one telco (fixed or mobile operator) to another, he is able to take his existing phone number with him. So although his phone number looks as if it belongs to one company (e.g. mobile operator A), in reality his calls should be routed to another company (mobile operator B). Most importantly, anyone making a phone call to this number should be charged according to the termination rates defined by B.

The only way to determine whether a number has been ported or not is to check it against the ported numbers database. These databases differ for each individual country, and have a variety of formats and access API structures. However, PortaBilling® offers a flexible method for handling LNP, using a system of number translation plugins that satisfy the requirements valid in various countries.

A basic plugin for LNP (where all ported numbers are imported into a local database on the PortaBilling® server) is supplied along with the system. When a call processing request arrives to PortaBilling®, the billing

engine checks if the original number dialed by the user is found on the list of ported numbers. If there is no match, call setup proceeds as usual. However, let us assume that a customer has dialed a number that originally belonged to the mobile operator Telenor, but was recently ported to another operator, Netcom. In this case, there will be an entry in the LNP table, and the ID of the phone number's new owner will be extracted (4792, in this example). This ID could be a symbolic name (e.g. Netcom), but usually it is more convenient to use a phone prefix or phone number belonging to that operator. The following will then happen:

- The billing engine looks up routing to a destination identical to the number owner's ID (4792, in this case), and the call is routed directly to the Netcom network.
- The same ID is used to calculate your termination costs and apply charges to your customer, i.e. he will be charged according to Netcom's rates.
- The originally dialed phone number is stored in the CDR information, but is linked to the destination used for charging the call. So the end-user will see "4791555123 – Norway, Netcom mobile" in the CDR report.

If a different way of extracting information from the LNP database is required for a given country (for instance, real-time lookup in a central database), you can create your own LNP plugin and use it with PortaBilling®.

Premium Number Billing

This is actually quite similar to the situation with ported numbers described above. PortaBilling® has to perform a lookup (either in a local database or an external one) to determine what the routing destination for the number is and which rating code should be used to charge your customer.

Favorite Number Billing

Although it represents a totally different business case, this type of billing uses the same internal architecture of number translation plugins, described in the previous section. The only difference is that, in this case, the dialed number is checked against a list of "favorite" numbers defined for each account. If a match is found, the call is rated according to a special rate for the **FAV** destination, defined in the customer's tariff. This allows you to offer customers a "call friends & family cheaper"-type service.

Active Call Monitoring

PortaBilling® can collect information about calls in progress from the PortaSIP® server, Cisco/Quintum gateways and other VoIP nodes. The only thing required on the VoIP node side is the capability to send Start accounting records, in addition to Stop records.

This enables several options:

- You may see a list of calls in progress on the PortaBilling® web interface.
- You may forcibly disconnect a specific call (if the call is routed via PortaSIP®).
- You may specify the maximum allowed number of simultaneous calls for each customer. PortaBilling® will reject any call attempts above the allowed number of concurrent calls.

IP Device Provisioning and Inventory

If you provide your VoIP customers with IP phone equipment, you know how laborious and yet important the task of performing initial configuration is. If the equipment is not configured properly, it will not work after being delivered to the customer. Or, even if it works initially, problems will arise if you need to change the IP address of the SIP server. How can you reconfigure thousands of devices that are already on the customer's premises? There are two ways to manage the device configuration.

Manual provisioning

The administrator must login to the device provisioning interface (typically HTTP) and change the required parameters. There are several drawbacks to this method:

- The IP phone must be connected to the Internet when the administrator is performing this operation.
- The administrator must know the device's IP address.
- The IP phone must be on the same LAN as the administrator, or on a public IP address (if the device is behind a NAT/firewall, the administrator will not be able to access it).

Due to these reasons, and since every device must be provisioned individually, this method is acceptable for a testing environment or small-scale service deployment, but totally inappropriate for ITSPs with thousands of IP phones around the world.

Auto-provisioning

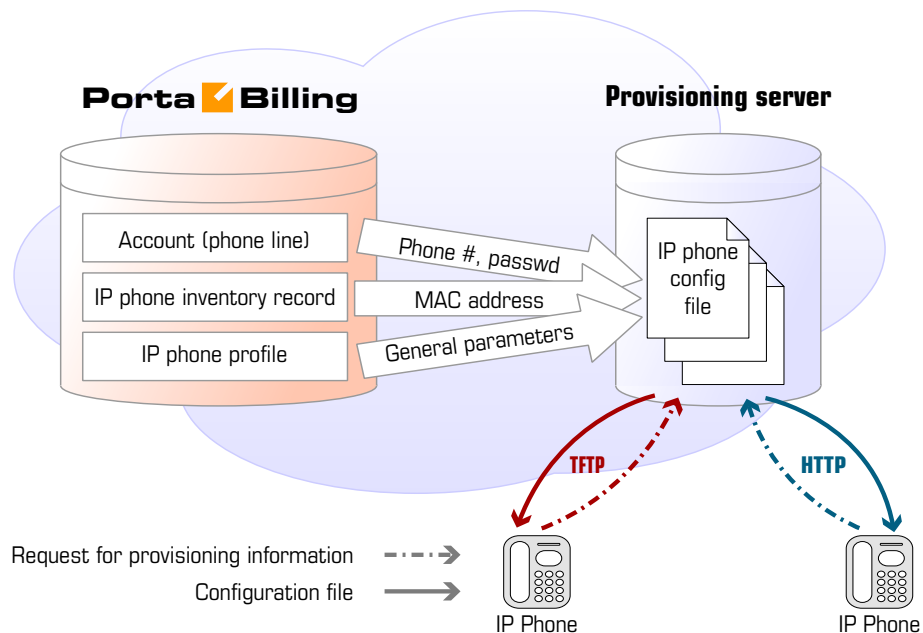
This approach is a fundamentally different one. Instead of attempting to contact an IP phone and change its parameters (pop method), the initiative is transferred to the IP phone itself. The device will periodically go to the provisioning server and fetch its configuration file.

IP Phone Provisioning

When you use auto-provisioning for an IP phone, instead of entering the same values for codec, server address, and so on into each of a thousand user agents, you can simply create a profile which describes all these parameters. Then PortaBilling® can automatically create a configuration file for the SIP phone and place it on the provisioning server.

The only configuration setting which is required on the IP phone side is the address of the provisioning server, i.e. where it should send a request for its configuration file. When the IP phone connects to the Internet, it will retrieve a specific configuration file for its MAC address from the TFTP or HTTP server and adjust its internal configuration.

If you decide later to change the address of the SIP server, you need only update it once in the profile, and new configuration files will be built for all user agents. Each user agent will then retrieve this file the next time it goes online.



The config file is specific to each user agent, as it contains information such as username and password; thus the user agent must retrieve its own designated config file. The following are defined in the billing configuration:

- The IP phone profile, so that the system knows which generic properties (e.g. preferred codec) to place in the configuration file.
- An entry about the specific IP phone in the IP phone inventory (including the device's MAC address), with a specific profile assigned to it.
- The IP phone (or, in the case of a multi-line device, a port on the phone) is assigned to a specific account in the billing.



Auto-provisioning will only work if your IP phone knows the address of your provisioning server. If you buy IP phones retail, you will probably have to change the address of the provisioning server on every phone manually. However, if you place a large enough order with a specific vendor, these settings can be pre-configured by him, so that you may deliver an IP phone directly to the end-user without even unwrapping it.

IP Phone Inventory

The IP phone directory allows you to keep track of IP devices (SIP phones or adaptors) which are distributed to your customers. The MAC address parameter is essential for every IP phone which is to be automatically provisioned, and so a corresponding entry must be created in the IP phone inventory.

Using Different Protocols (SIP and H323)

The billing itself is independent from the actual implementation details of those protocols used to carry out the call. PortaBilling® communicates with the nodes on your VoIP network (gateways, gatekeepers, proxies) using the RADIUS protocol and the VoIP VSA extensions for it. Thus, from the point of view of PortaBilling®, RADIUS requests related to SIP calls are nearly identical to those related to H323 calls (there are only a few different attributes).

PortaBilling® is able to process both types of requests, so no special configuration in billing is required to support SIP or H323 calls. You only need to add your gateway to the system as a trusted node, and then both H323 and SIP calls reported from that gateway will be processed properly.

Note that from the billing point of view, a SIP server is not different from any other gateway - it also issues authentication or authorization requests, and sends accounting information after the call is completed.

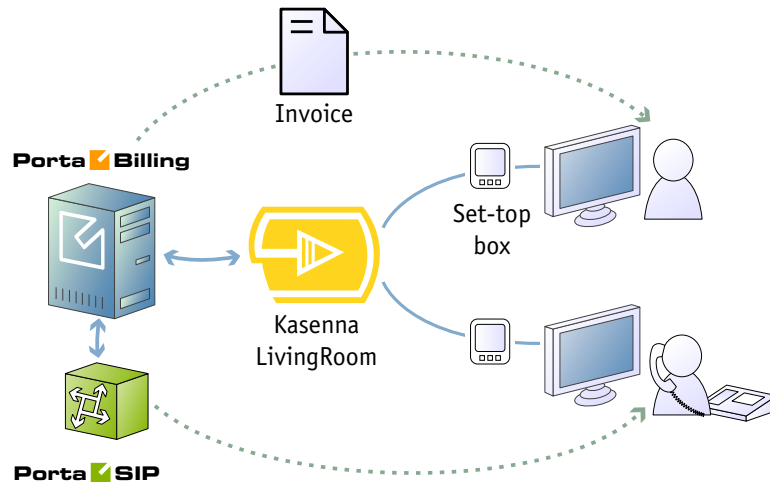
A node for the SIP server should therefore be created in your system, so that it can communicate properly with the billing. Also, products that are

to be allowed to use SIP services should have an Accessibility entry which includes this node.

5. Converged Services

IPTV

For today’s service providers, “triple play” is not just a buzzword - rather, it is the only way to stay competitive and retain their customer base. Combined solution provided by Kasenna and PortaOne can be used to add IPTV to your service portfolio and expand your business growth.



Kasenna PortalTV includes a content delivery platform, a services framework, application servers, and client portals. Together they enable the acquisition, storage, distribution, caching and delivery of content to a variety of devices; the management of content workflow; rights management and royalty reporting to rights owners; and the delivery of IPTV content to various applications (such as set-top boxes).

PortaBilling is a converged real-time billing system, service provisioning environment and CRM used for rating and invoicing any type of service you provide to customers, with centralized funds management and consolidated bills.

The administrator can define multiple subscription packages for IPTV (just like for any other service, such as internet telephony, messaging, or data transfer). These are mapped into channel bundles in the LivingRoom. This allows you to do better marketing, work with multiple customer groups, and enjoy greater flexibility in terms of pricing the service.

When a new or existing customer signs up for the IPTV service, the available channel information (based on the assigned subscriptions) is provisioned into LivingRoom via an extensible XML API. Also, if a customer's product configuration changes, or a customer is blocked (e.g. due to non-payment), this information is transferred to the LivingRoom.

If a pay-per-view service is provided, LivingRoom offers a framework for adapters that communicate with billing. So when a customer requests pay-per-view content, it can be validated in real time whether he has enough funds available, and the given amount can be debited from his account immediately after he confirms the purchase.

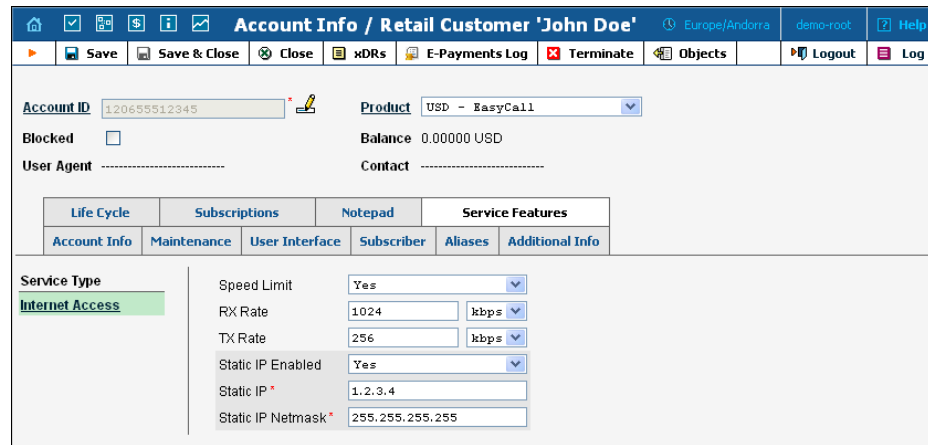
As a result, both subscription-based and pay-per-view IPTV services are fully managed via the PortaBilling web interface, and all applicable charges are added to the customer's invoice, along with other charges (periodic and event-based) for services such as internet access, voice calls, and so on. This enhances the customer's experience, improves accounts receivable tracking, simplifies maintenance, and reduces operating costs.

The combination of Kasenna PortalTV and PortaOne PortaBilling creates a world-class video ingest, storage, distribution and delivery platform with a robust, high-quality video functionality. Easy provisioning and full control on the billing side allow you to reduce operating costs and quickly achieve profits from IPTV services.

Internet Access Service

Provisioning

PortaBilling® allows you to manage the provisioning of Internet access services when a user establishes a dialup session with your NAS (Network Access Server) for further Internet access. PortaBilling® performs the session authorization (validating that this user is allowed to use this service and what the maximum allowed session time is) and, when the connection is terminated, deducts the cost from the customer's balance according to the session time and the specified rate (for the special destination **DIALUP**). A PortaBilling administrator can provision the amount of available bandwidth per product or individual account and also perform static IP allocation for certain accounts. All these settings are then returned to the access server during authentication, and so will be applied to the session.



The screenshot shows the 'Account Info / Retail Customer 'John Doe'' page. At the top, there is a navigation bar with icons for home, save, save & close, close, xDRs, E-Payments Log, terminate, objects, logout, and log. Below this, the account details are displayed: Account ID (120655512345), Product (USD - EasyCall), Blocked status (unchecked), Balance (0.00000 USD), User Agent, and Contact. A menu bar includes Life Cycle, Subscriptions, Notepad, and Service Features. Under Service Features, there are tabs for Account Info, Maintenance, User Interface, Subscriber, Aliases, and Additional Info. The 'Service Type' section is expanded to show 'Internet Access' configuration: Speed Limit (Yes), RX Rate (1024 kbps), TX Rate (256 kbps), Static IP Enabled (Yes), Static IP (1.2.3.4), and Static IP Netmask (255.255.255.255).

PortaBilling® can be used as a back-end for any NAS which uses a Cisco-compatible RADIUS protocol.

Controlling Bandwidth

PortaBilling allows you to control the amount of bandwidth provided to a customer. There are two available methods:

- PortaBilling can supply specific values for the upload and download speed during authorization. This is the easiest method to use. The disadvantage here is that the same bandwidth limit will be used for the entire duration of the session.
- Alternatively, PortaBilling can return the name of a policy to the NAS, and bandwidth allocation will be performed in the NAS based on the definition for that policy. The advantage of this method is that you can use advanced options in the policy definition, e.g. providing different limits during the day and at night. Please refer to [PortaSwitch Handbook: Converged Services](#) for examples of policy configuration on Cisco routers.

Rating Data

Since Internet access is an “always-on” service, billing is typically done in real time, based on RADIUS update (“keep-alive”) requests received from the NAS; thus many keep-alive requests may be combined into a single xDR. In order to provide detailed information on the charges applied, the contents of each update request are stored in the PortaBilling internal tables, and both summaries and detailed data can be browsed on the PortaBilling web interface.

Data Transfer Quotas

Data transfer quotas function in a similar way to volume discounts for services such as voice calls. You can define combinations of data transfer

thresholds, along with the applicable discounts, e.g. the first 5GB free (100% discount), after which the customer is charged the normal rate (\$0.01/MB).

You also have the ability to adjust the service itself once a threshold is reached, e.g. when the customer exceeds 5GB his download speed will be severely reduced (but he can still use the web, email and the like). If he continues to use the service in an abusive manner, and so exceeds 6GB, the service will be automatically blocked.

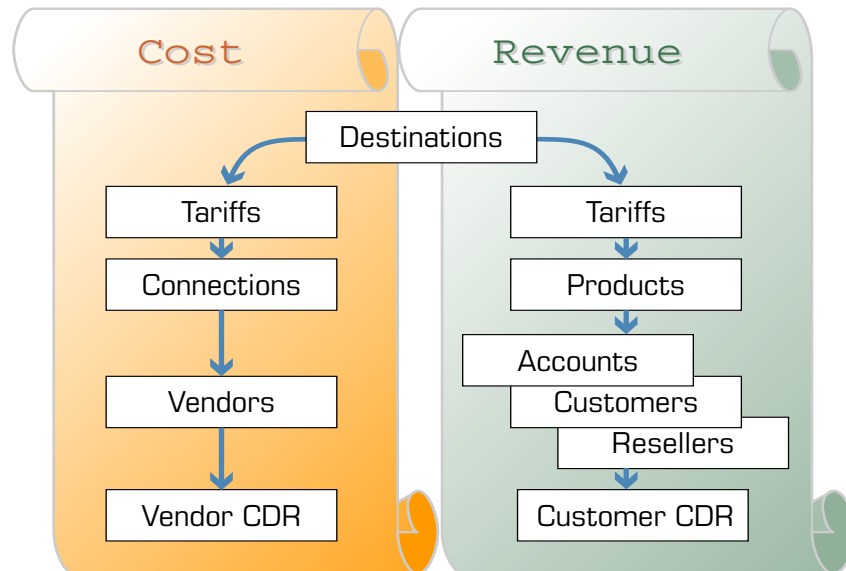
Service Shutdown

PortaBilling can initiate the disconnection of a session in progress on NAS; for instance, if it is determined that the customer has exceeded the allowed data transfer limit, or if the customer has been suspended due to non-payment. In this case, PortaBilling sends NAS a POD (packet of disconnect).

6. Entering Data in PortaBilling®100

Cost/Revenue

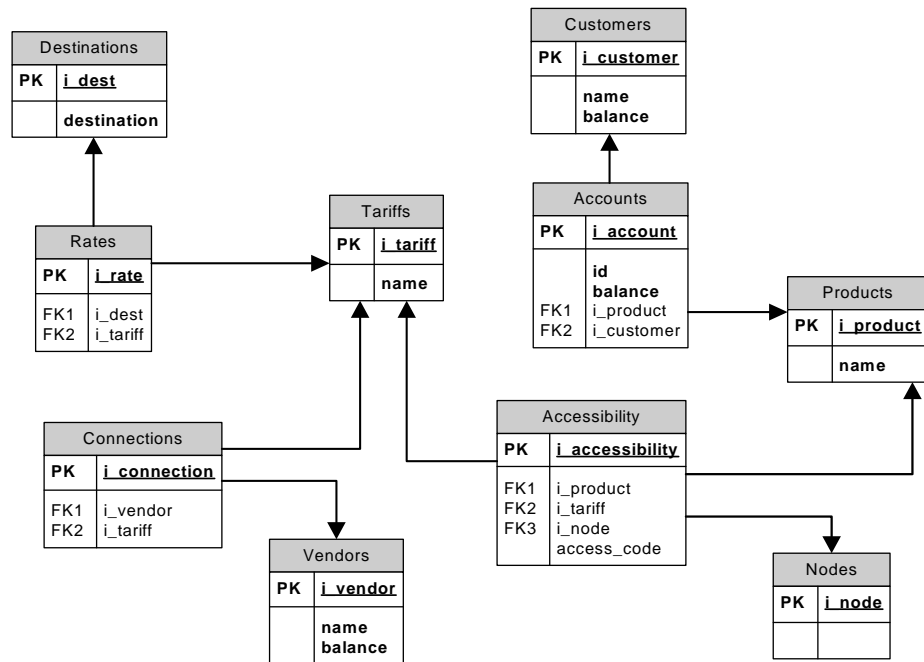
There are two independent parts of the billing configuration:



The Cost part provides you with information about expenses associated with running your VoIP business: call termination, incoming toll-free lines, etc. The Revenue part calculates charges applicable to your customers. Note, that these parts are independent (the only common item is Destinations). So, the amount charged to your customer is unrelated to your termination expenses and vice versa.

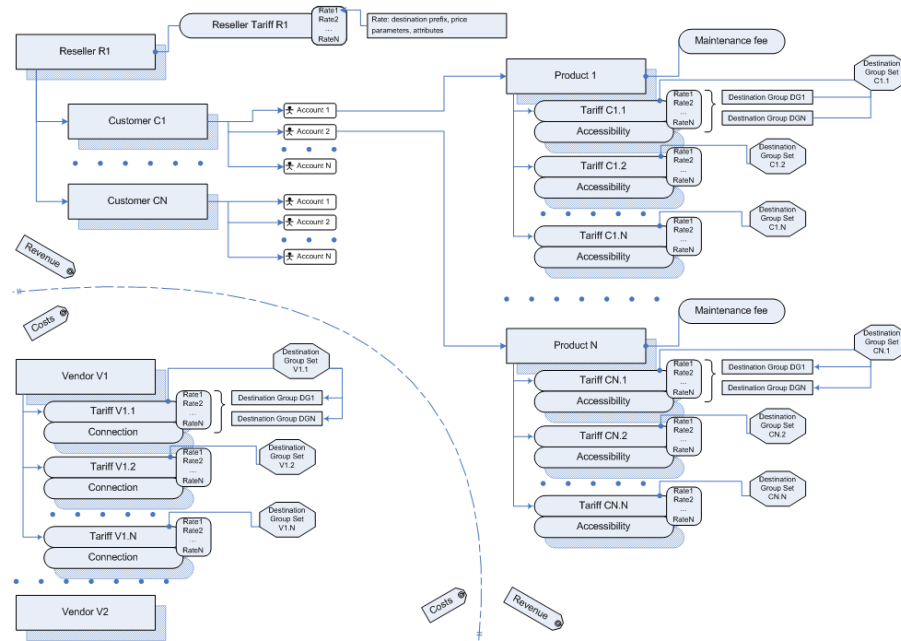
PortaBilling® 100 Data Model Concept

The illustration below shows some major elements and their interrelationships. For the sake of clarity, only the most important tables and columns are shown.



- Destinations define possible phone number prefixes which will be used in the system.
- Rate tables store information about rating parameters for a specific destination with a specific tariff.
- Tariffs serve as a pricelist to be assigned to a specific customer or vendor
- The relationship between vendors and tariffs is defined via connections. A tariff assigned to a specific connection will be used to calculate termination costs for a given vendor.
- Every account is assigned a certain product, which defines how the account will use the service and how it will be charged for this.
- A product is related to tariffs via an accessibility table, so that a single product may be linked with multiple tariffs.
- Accessibility specifies which tariff is applied to the account for calls made on a specific node (gateway) and access number.
- Accounts are linked to a customer, so that a single customer can have multiple accounts.

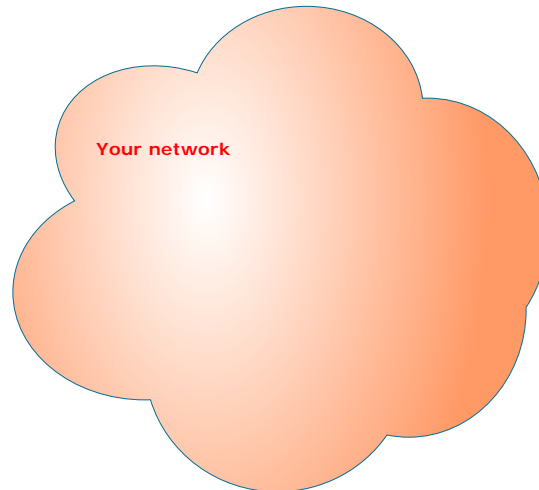
Billing Entities Hierarchy



The diagram above presents a hierarchical overview of the billing information stored in the system. On the cost side, under vendors, are located connections, with a tariff associated with each connection. On the revenue side, accounts are grouped under a customer (while a customer can, in turn, be placed under a reseller). The account is linked with the tariff (rate plan) via its product; note that this relationship is “one to many”, so that different tariffs can potentially be applied to the same account.

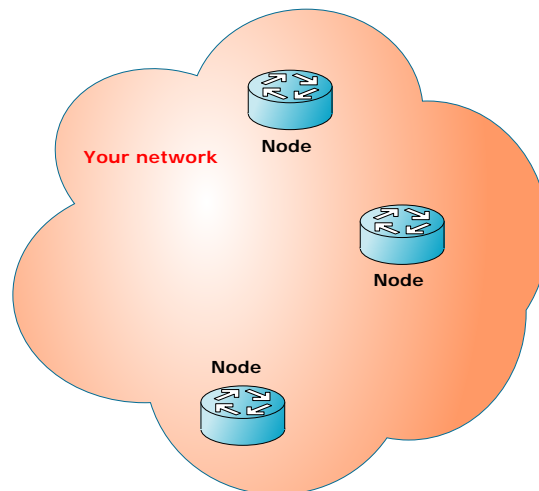
Entering Data

Step 1 – “Your Network”



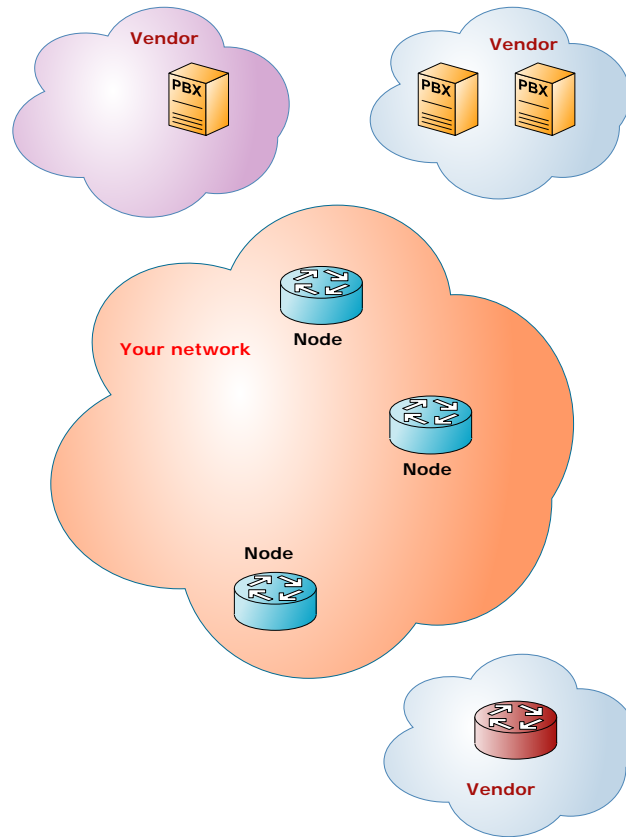
PortaBilling® is designed for an IP telephony service provider which we will call the **Company**. All information kept within the system is used to calculate the financial status between the Company and its **Vendors** and **Resellers** or **Direct Customers**.

Step 2 – “Nodes”



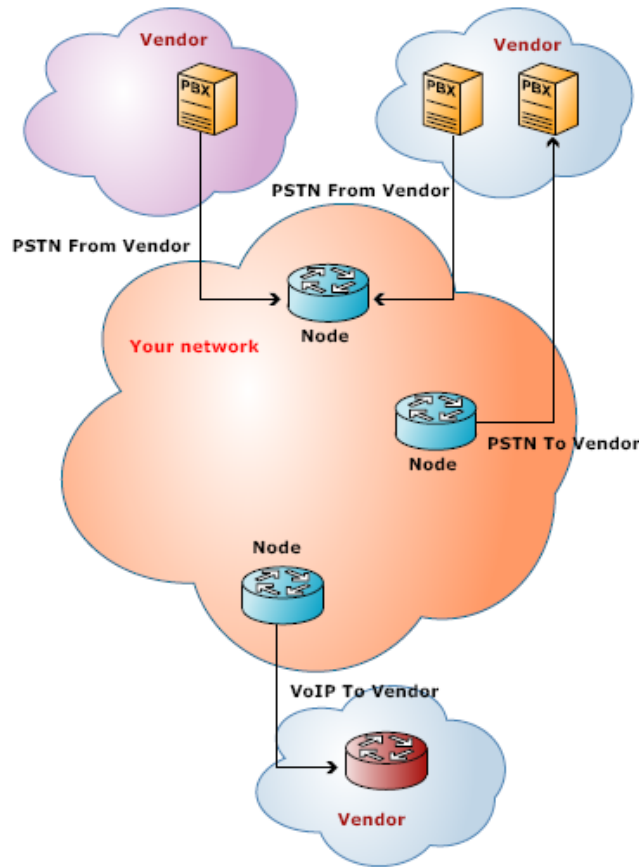
The **Network** is defined as all network elements, or **Nodes**, that belong to the Company. Gateways in the Company’s network are “trusted” gateways. These gateways must be listed in the Nodes registry, and are also required to be Radius clients so that PortaBilling® may receive accounting information.

Step 3 - "Vendors and Customers"



Vendors are those partners which provide VoIP traffic termination services to the Company. If the Company sends traffic to another service provider, then, in this case, the other entity is defined as a Vendor. Therefore, if there is bilateral traffic between the Company and another firm, the second firm must be registered as both a **Vendor** and a **Customer**.

Step 4 – “Connections”



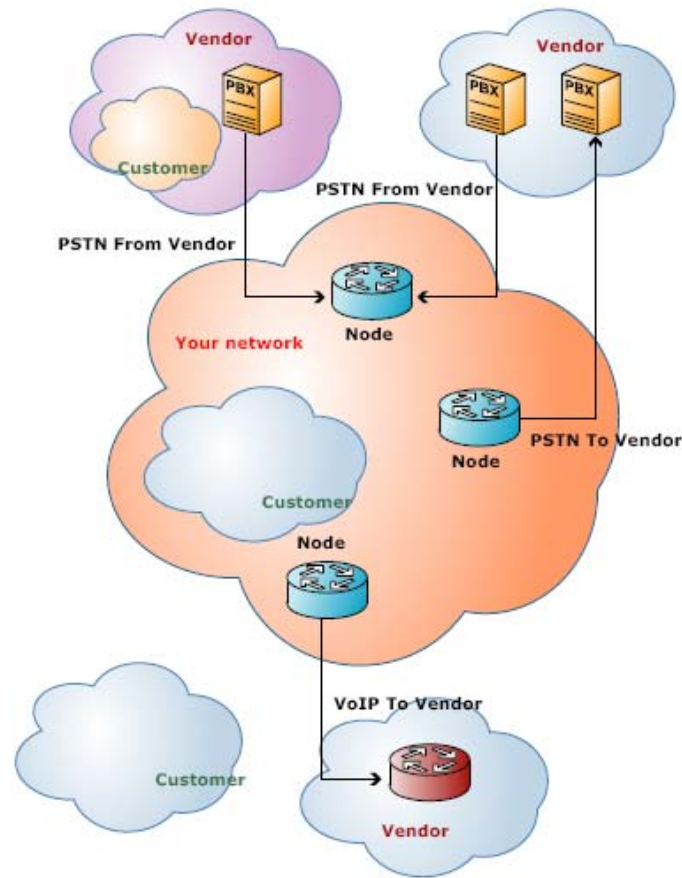
A **Connection** is a precise description of a point of contact between the Company’s Network and the Vendor’s. It is the point at which costing occurs and revenue is generated. There may be multiple Connections to the same Vendor, and each may, for example, have a different price. The different types of Connections are categorized as

- “VoIP to Vendor”, “PSTN to Vendor” – outgoing calls
- “VoIP from Vendor”, “PSTN from Vendor” – incoming calls

Sometimes two entities are confused:

- The vendor who delivers calls from **your** customers to your network (providing you with a toll-free line or DIDs) – this is a VoIP or PSTN from vendor connection
- The vendor who sends calls from **his** customers to your network for termination – in this case he is just another one of your customers, and not a vendor

Step 5 – “Vendors and Customers Part II”

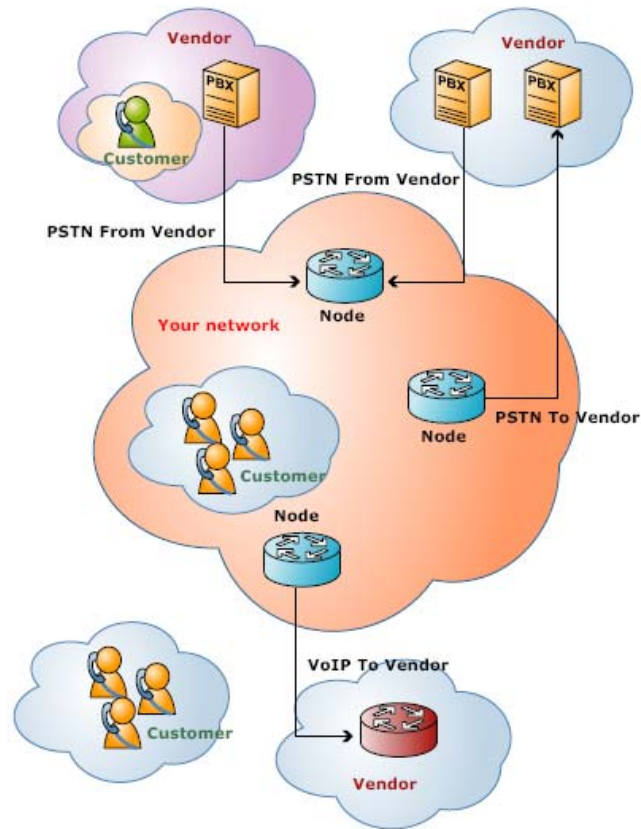


Customers are logical entities in the system which will be charged for the calls they make, and for whom invoices will be generated. They may be end-users of the service, resellers of a service, or a virtual customer used to keep track of prepaid cards distributed by the Company.

A **Retail Customer** is an abstraction which groups a number of Accounts together. This could be, for example, a company with an account for each employee. The company will be invoiced for all phone usage, and PortaBilling® will charge it for this total, with the company being able to see details of usage for each account. Retail customers may belong directly to the owner of the system (direct customers) or to a Reseller.

A **Reseller** may be the reseller of a service. In this case, the Reseller is charged according to the Reseller’s Tariff, but accounts sold by the Reseller could be charged using different Tariffs (Products), managed by the Reseller.

Step 6 - "Accounts"



A Customer is composed of one or more **Accounts**. These Accounts may represent individual users of a product, such as prepaid cards or IP phones; or they may identify remote gateways.

7. Management and Monitoring Tools

Templates

Templates are a powerful and flexible tool defining the relationship between data and its representation. Let's take a practical example: vendors send you their rates, and you have to enter this data into the system. Obviously, regardless of the particular numbering plans, formats and other options, this data has to be in the unified format used in PortaBilling®. Some vendors use Microsoft Excel, others text-based formats (for example, .CSV). Even if the same format is used, there will be different ways of positioning and formatting the data. For example, vendor A might have the phone prefix for the Czech Republic in the first column in his data file, entered as **420**. Vendor B might have it in column number three, entered as **011420**.

Of course, it is possible to write an external tool which would convert data files from these different formats into a common one. However, this would require some extra steps in data processing, so that every time you upload rates from a vendor you will have to perform a conversion first. In addition, this will probably involve extra cost, since not all users are capable of writing the required data conversion application.

This is why PortaBilling® has chosen a different approach. Instead of making your data fit PortaBilling®, you can make PortaBilling® understand and process your data as is! The PortaBilling® rate upload wizard and templates allow you to do the following:

- Upload your XLS or CSV rate file to the server and immediately see the result of file processing (in order to verify that the file was recognized correctly, and to adjust parameters such as the field delimiter).
- Using a drag-and-drop interface, you can place markers on columns in the file to identify where particular groups of data are located in it (e.g. the destination prefix in the third column, and the price in the fifth column). In fact, the rate upload wizard will try to automatically recognize data elements, so most of the time you will only need to confirm the proposed column assignment.
- You can adjust the format which each data element is in (for example, whether the phone prefix is given as **011420** or as **420**). You can also apply post-processing rules (analogous to formulas in spreadsheet editors).
- The upload wizard will process all the rate information and allow you to compare new rates with the existing rates and make adjustments if necessary.
- If there are new destination prefixes in the file (not in the database yet), they can be created automatically during rate upload. Also,

you can assign new prefixes into specific destination groups during the upload process.

- Finally, when the new rates are uploaded into the system PortaBilling® saves all the changes you made as a template associated with this tariff. So next time you receive a new file from the vendor and need to upload it, everything is ready and no special actions are required.

For more details about templates, see the [PortaBilling Templates Guide](#).

Date and Time Information

Time Zones

Due to the global nature of the VoIP business, it is very important to identify times with reference to time zones. For example, there is an 18-hour difference between 8pm Melbourne, Australia, and 8pm Seattle, USA – which could mean many thousands of calls and thousands of dollars incorrectly counted if the time zone is not taken into consideration.

This is why every object in PortaBilling® (e.g. user, customer, vendor or account) has a time zone associated with it. All dates are stored in the database in a universal, portable representation, and are converted into the required time zone when necessary. For example, say a call is made at 14:03 New York time (the time zone of your PortaBilling® server). The call is made using a calling card running on the America/Chicago time zone, and the account belongs to customer Easy-Cards-Reseller, whose time zone is America/Los_Angeles. The call goes to vendor Deutsche Telekom, which has Europe/Berlin as its time zone.

Thus, if each of these entities views information about the call, they will see it in their respective time zone, as follows:

- Account: 13:03
- Customer: 11:03
- Vendor: 20:03

This helps to present time information in a format suitable for the user. The time zone also affects the billing period. Thus if a customer has time zone Europe/Berlin and a monthly billing period, this means that his billing period covers one whole month, from 00:00 on the first day of the month until 00:00 on the first day of the next month **in his time zone** – Western European Time, in this example.

Also note that all time zones are expressed in relation to a geographical location in PortaBilling®, for example, Europe/Prague. This helps to avoid ambiguous abbreviations such as EST, which could mean both New York time and Melbourne time. Moreover, by using such a notation daylight savings time automatically comes into effect when applicable, so you do not have to think about whether it is EST (Eastern Standard Time) or EDT (Eastern Daylight Savings Time).

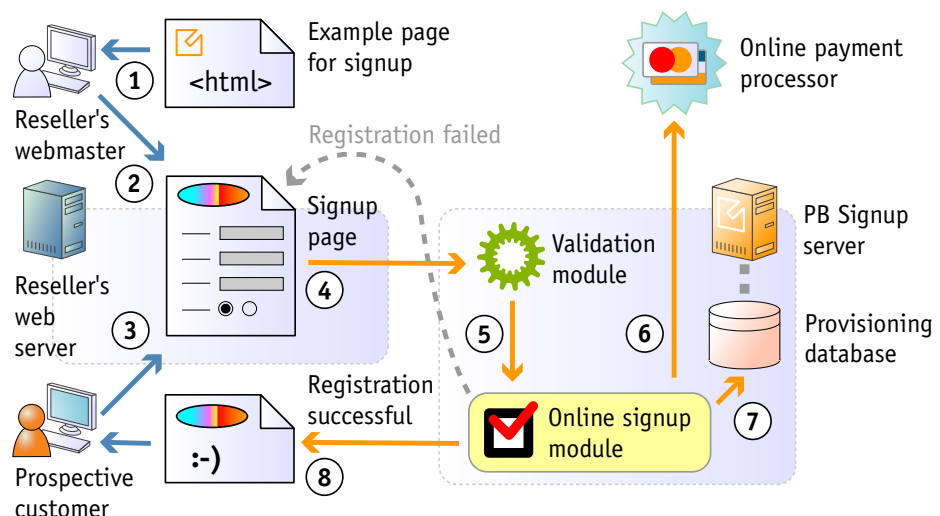
Date and Time Formats

Times and dates have different formats in different places in the world. For example, the same date will be represented as 20.05.2003 in Germany and 05/20/2003 in the US. It is important that the user be able to work with dates in his native format in order to facilitate work and avoid data interpretation problems.

Each PortaBilling® user (admin, account, customer care) has his own date and time formats. When logging in to the web interface, all date and time information will be shown in the user's specified format. Additionally, all data input fields will also assume the date and time formats specified for the given user.

Online Web Signup

PortaBilling® allows you to automate the process of subscribing new customers to the service, so that customers can do everything necessary online without requiring any assistance from your administrator. A template-based API for online signup allows you to customize page layout and put a web signup front-end on multiple websites (web portals, web pages of your resellers or affiliates, etc.). The following diagram illustrates the online web signup process:



A fully functional web signup portal is supplied with PortaBilling®. You can use it out of the box, or take it as the basis for further customizations.

Web signup allows you to:

- Collect customer name and address information;
- Allow customers to select from the available phone numbers;
- Choose a desired product bundle;
- Collect a customer's credit card information;
- Authorize credit card payment, fully provision the account information in PortaBilling®, and provide service-related information (e.g. auto-configuration file for a softphone).

This is a three-level application which allows you to deploy multiple online web signup sites without compromising security, and to easily customize each site to fit a specific business case.

- Front-end – These are the web pages customers will access using their web browser. It is a set of HTML pages (with Javascript and CSS functionality), which can be fully customized in terms of page layout, colors, etc.
- Middleware – This is the code located on the PortaBilling® server or another secure server. It retrieves the required data from PortaBilling® using XML API and communicates with the front-end using AJAX. This serves as a protection layer against any malicious web requests.
- Back-end – Located on the PortaBilling® server, it performs credit card authorization and actual data modification in the database.

Custom Reports

If your business process requires that specific information from PortaBilling® be presented in a certain format (e.g. your marketing department needs to know the top 10 countries based solely on calls longer than 15 minutes, including the ratio between these calls and all other calls for each country), and this information is not available among PortaBilling®'s default reports, then there are two options available:

- Create your own custom report using tools such as Crystal Reports (for more details, see the *How to ...* section). This assumes that you have sufficient experience with relational databases and reporting tools.
- Contact the PortaOne team, so that a custom report can be developed for you and uploaded to your system. In this case, no technical knowledge on your side is required.

PortaBilling®'s custom report module allows the creation of “report” objects, which can later be uploaded to your system and then executed by your administrators. In this way, you can maintain an unlimited number of different reports specific to your business.

Report type

Report type defines the purpose of a report: does it show the most popular destinations, or the customers who make the most calls? Several types of reports are already supplied with PortaBilling®, including:

- Most popular destinations of a specific customer
- Total amount of minutes and charges for a specific customer

Other types of reports can be developed on request. When a new report object is uploaded to your system, a new type of report will appear.

Report type defines:

- What types of input parameters are required by the report (e.g. customer name, time interval, etc.).
- What the output values of this report are. Since the report results are displayed as a table, this defines what the columns represent in this table, while every row in the table represents another line of report data.

Report query

Whereas report type defines a report's general characteristics (e.g. “Most popular destinations”), by executing a report query you may obtain a specific result (e.g. “What were the most popular destinations for customer John in the last two weeks?”).

Sometimes you will simply execute a query once (for instance, you would like to quickly verify that a particular customer did indeed make more than 1,000 calls yesterday), while other types of queries will be made routinely (e.g. every morning your sales manager checks the previous day's statistics for his VIP customers). To accommodate both types of needs, PortaBilling® allows you to create a new query and then:

- Run it once in your web browser to see the results immediately.
- Save the query (including input parameters) for later use. When you need to re-run the query sometime in the future, you simply open and execute it, i.e. you do not have to fill in any parameters. This significantly reduces the time required to prepare complicated reports, and allows you to minimize the chance of error. The administrator creates the query just once, and then inexperienced users need only execute it.

Also, to save you time when preparing multiple similar reports, PortaBilling® allows you to create a query as a copy of another existing query. After you have defined a complex report with many input parameters for customer A, simply create a copy of it and change the

customer name to “B”. Now you have reports for both customers, either of which can be executed with just one mouse click.

Query execution

There are several methods for executing a query and obtaining report results:

- You may request query execution from the PortaBilling® web interface, and see the result in your web browser.
- You may request that a query be executed in the background, with the results delivered to you via email. This is the preferred method for complicated reports involving large arrays of data, since in this case PortaBilling® can schedule query execution in such a way that it does not interfere with other tasks. This allows you to avoid performance degradation when several users are trying to execute reports concurrently; PortaBilling® will execute all these reports sequentially.



This is the recommended method for executing reports.

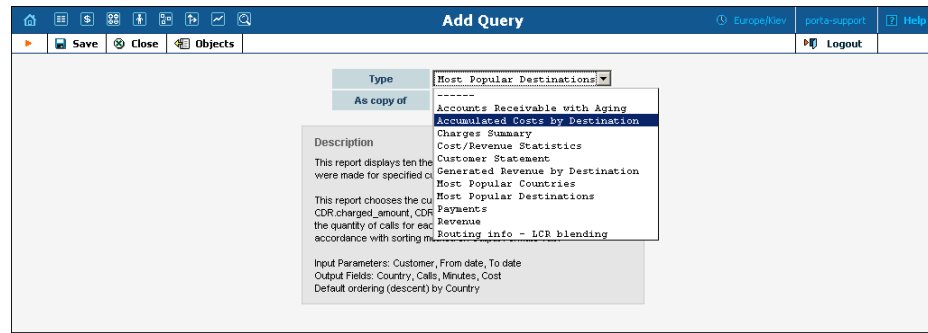
- You may schedule a query to be executed at a certain moment in time or periodically, with the report results delivered to you via email. This allows you to automate reports that you use on a continuous basis. For instance, you may schedule a “Total minutes per customer” report for 3am every day; then every time you open your mailbox in the morning, you will find all the required information. You can view a list of scheduled query executions and their status on the PortaBilling® admin interface.

Query results

When you execute a report query from the web interface, its result is stored in PortaBilling®. Thus if you did not save the results initially, you need not run the report again, as the data for the last ten report executions is always available.

Custom reports supplied by default

PortaBilling® includes a comprehensive set of reports, which are already included in the system upon initial installation:

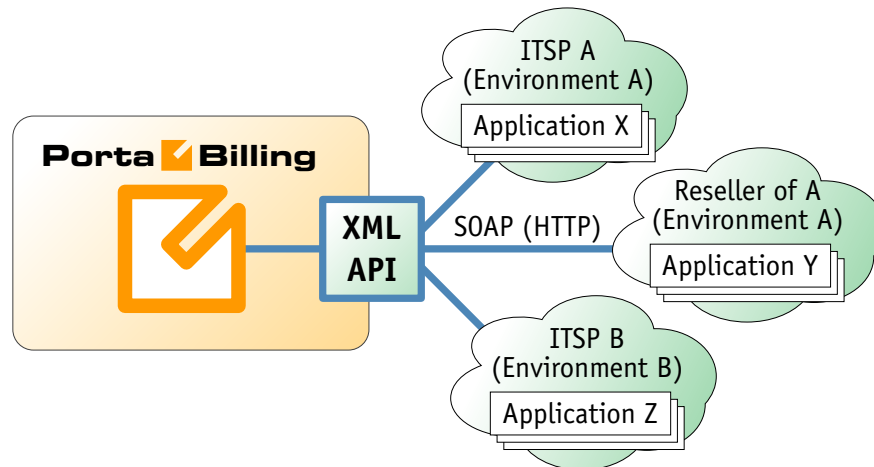


Report	Purpose
Most Popular Countries	Allows you to monitor which countries particular customers call most often.
Most Popular Destinations	Allows you to monitor which destinations (individual prefixes) particular customers call most often.
Payments	Provides details about payments recorded in the system during a specific time interval (with the option of filtering payments for particular customers).
Customer Statement	Prints out a list of a customer's invoices / payments with the running balance for a given time interval.
Revenue	Shows the total revenue (all the charges to customers) during a specific period. It includes the following report types: <ul style="list-style-type: none"> • Revenue by date • Revenue by charge type • Revenue by customer class
Accounts Receivable with Aging	Shows the sum of invoices not paid by customers (per customer and per periods 0-30, 31-60, 61-90 and >90 days).
Generated Revenue by Destination	Shows generated revenue for specific services and customers grouped by destination.
Accumulated Costs by Destinations	Shows accumulated costs of a service for a specified vendor grouped by destination.
Charges Summary	Allows you to obtain a monthly report about a specific reseller's customers and their charged amounts in the form of a single list (CSV file format). This makes it convenient for subsequent computer processing (typically used by a Reseller). The main purpose is to relieve the Reseller from having to conduct endless customer searches.

Routing info - LCR blending	Analyzes and shows the available routes, calculates the average termination price for each destination, and then uses this as a basis for computing the rate for a given customer. The most important step is to “blend” the prices of different vendors according to a realistic scenario. For more detailed information, see the section Routing Info – LCR Blending of Wholesale Services Handbook .
Cost / Revenue Statistics	Allows you to find each customer traffic pattern through vendors per destination. Also shows ASR, ALOC, cost and other parameters for any given period of time.

XML API for Data Operations

In addition to the ability to access billing data directly in the database, PortaBilling® allows you to perform operations such as data retrieval or data modification via XML API.



This method has several advantages:

- It is based on SOAP (Simple Object Access Protocol) and HTTPS transport, so it is accessible from any platform or operating system, and all communication between the server and clients is secure.
- The business logic embedded into the API provides integrity checks for all data modifications, as well as data composition for data retrieval.
- XML API is accessible by every owner of a virtual environment or reseller. Each user’s access is automatically limited to his “visible” portion of the available data, e.g. a reseller can only retrieve information about his own sub-customers or their accounts.

XML API allows users to perform select, update, insert or delete operations on entities such as customers or accounts. Each user has his own login credentials, and each operation he wishes to perform is analyzed to determine if it is possible with regard to general data integrity (e.g. a new account cannot be created without being assigned to a customer) as well as the particular user's security permissions (ACLs) (e.g. while it is possible in general to create new accounts, this user may be prohibited from doing so).

Trouble Ticketing System

In order to meet customers' expectations for quality of service, it is important to keep detailed information about problems which they currently have or had in the past. In addition to information about the service configuration and rating (stored in PortaBilling), you must also keep track of issues reported by a particular customer and the full history of communication and comments on each issue. This allows you to re-use information obtained earlier about the problem (one of the most frustrating things for customers is being asked the same questions all over again), receive important analytical and reporting data, detect patterns of how problems emerge, and re-use troubleshooting methods successfully applied in the past.

This function is performed by a trouble ticketing system, namely, Best Practical's RT. This is a robust enterprise-grade ticketing system used by many companies worldwide (PortaOne, Inc is one of them). You can find more details about the architecture, functionality and features of RT on Best Practical's [website](#).

RT provides the ability to keep information about individual incidents (tickets):

- Each ticket is associated with a customer and has a unique ID (ticket number) and status (e.g. new, open, resolved). There are some other ticket properties (e.g. the administrator to whom this ticket is currently assigned), but their use generally depends on how the internal ITSP and customer service processes are organized.
- Each ticket has one or more requestors assigned to it. A requestor is a person who communicates with a helpdesk representative. When the helpdesk representative posts new information to the ticket via a **reply**, it is received by all requestors for that ticket via email. (Customer service may also exchange **comments**. These are for internal use only, and are visible only to helpdesk staff, not to requestors.)
- Incoming email communications from administrators or requestors are automatically recorded to the corresponding ticket

and then re-sent to the other parties subscribed to that ticket. It is possible for administrators to review the full ticket history from the web interface, while requestors can view the status of tickets and their contents from the web interface (obviously, they can only see **reply** messages). The web interface can also be used to post new messages to the tickets.

PortaBilling is fully integrated with RT, making it a component part of the overall CRM solution. This means that although RT is installed on a separate server and maintains its own database, there is no need to duplicate the entry of any information or log in into each system separately. Information about administrators (helpdesk) and customers is automatically propagated to RT, while information about a particular customer's tickets is dynamically retrieved from RT and presented on the PortaBilling web interface. Thus information about an existing ticket may be retrieved, or a new ticket created, with a single mouse click.

Billing Server Health Monitoring

Entire System Load

These statistics are accessible from the main admin menu via the **System load** link in the **Statistics** section.

This graph shows the number of calls registered by a production system every 15 minutes: normal calls with duration >0 (green areas) and calls with zero duration (red line). The most recent information appearing on the right-hand side of the graph is an hour old or less.

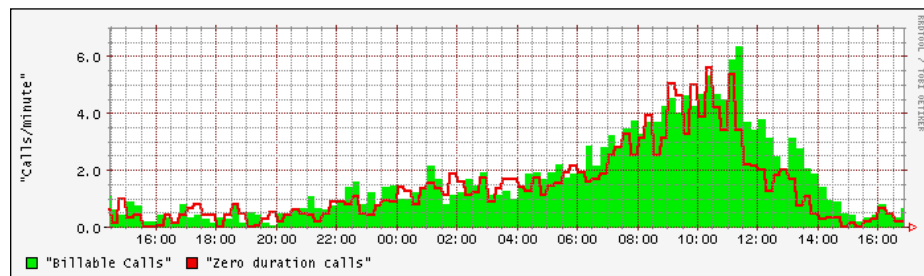


Figure 7-1

Number of calls

If the disconnect time on the CDR does not fall within the past 15 minutes, the call may not be finished, and hence will not be reflected on the graph. (See picture below, call types 3 and 4.)

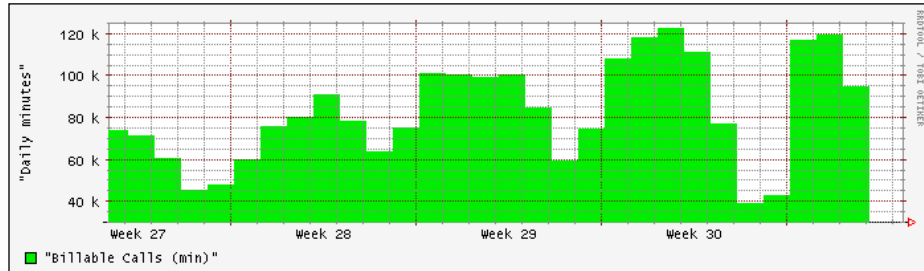
Zero duration calls

The graph can also show possible problems in the system, such as an unexpectedly high number of failed calls.

By default, this graph displays statistics for the last 30 hours.

Total Minutes Statistics

The graph below gives you the ability to monitor how the call volume in your system changes each day.



Master and Slave MySQL Server Statistics

These statistics are accessible from the main admin menu via the **Database** link in the **Statistics** section.

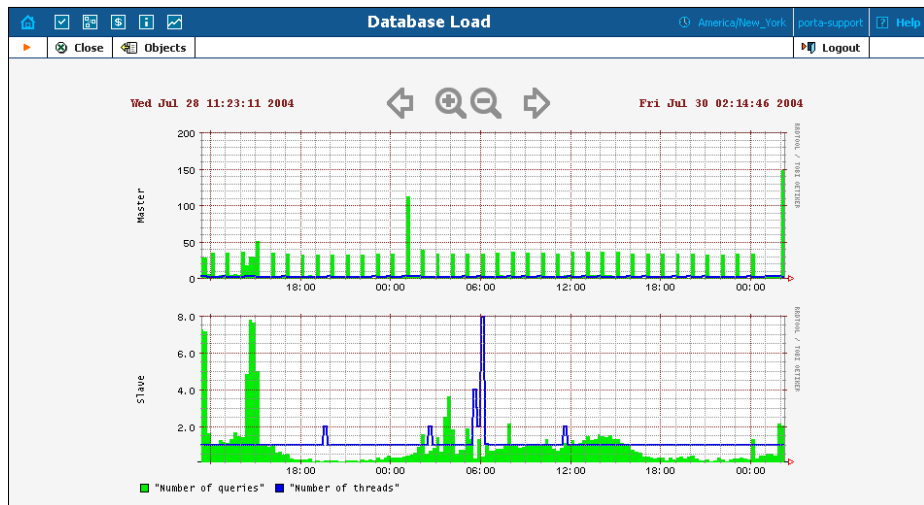


Figure 7-2

These two graphs show values for the status of the MySQL servers. The green line indicates the number of queries processed by the server every 15 minutes, while the blue one shows the number of threads running on the server.

Queries

The graph indicates how many times clients have queried the database. On recommended hardware, this value may exceed 200 without any difficulties. High peaks with a number of requests over 2,000 could indicate improper configuration of the system or temporary problems. Also, these peaks can appear when replication has been restored after a system fault.

Threads

Normal values for these graphs are:

- for the Master server: <=2
- for the Slave server: >=1 and <=3

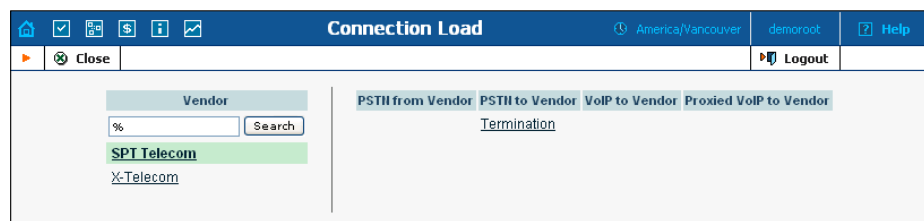
Normally, these values will depend on the processes running on the server, such as the Radius daemon, the Apache httpd server, statistics collection tools, and so on.

Larger values can indicate other client connections to the server. If the number of threads on the slave name server is 0, this means the replication process is down and requires administrator attention (see section 2.4 for details).

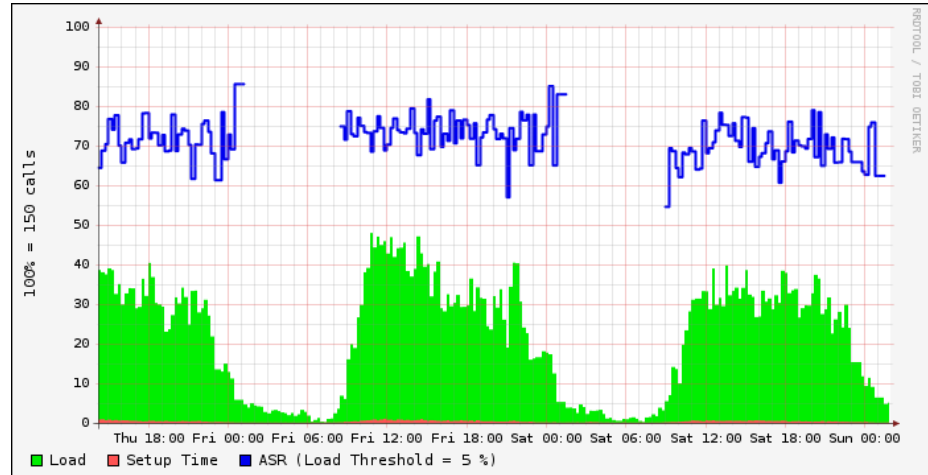
VoIP Network Performance Statistics

Connection Load

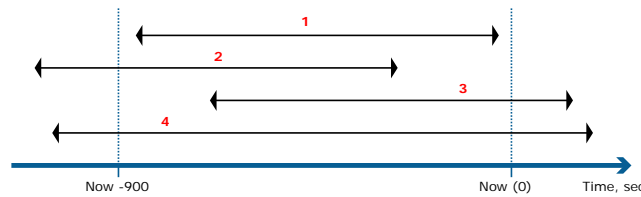
You can monitor load and quality parameters for individual connections. To see a list of the available connections, follow the **Connections** link from the **Statistics** section in the main menu.



After you select a vendor and a particular connection, you will see its load graph:



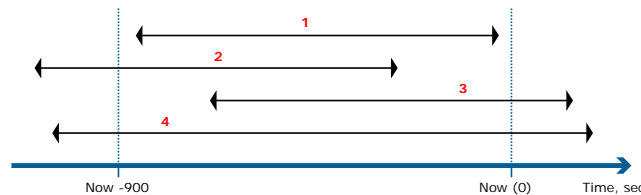
This graph shows the node load, setup time and ASR (Average Success Rate) statistics. The green area indicates node load, the red shows relative setup time, and the blue line indicates ASR. Information appears on this graph with a one hour delay.



Assumptions: To make load calculations easier, we ignore calls longer than one hour (since these are quite rare, it should not affect the final results much). Also, although a failed call has “zero” duration, it still occupies a port on the gateway for some time (while the gateway is trying to connect the call). Therefore, when calculating, failed calls are regarded as occupying the port for 10 seconds.

Connection load calculations

For a better understanding of the calculations performed, consider the following diagram:



The total duration must be calculated for all calls processed by the system corresponding to the interval between “now - 1 hour 15 minutes”, and “now - 1 hour”.

Four different call types are recognized in the above diagram; an additional call type is “zero duration”:

1. Sum of durations of all the calls.
2. Sum of durations between `t0` and `disconnect_time`.
3. Sum of durations between `connect_time` and `t1`.
4. Calculate sum as number of calls * 900 seconds.
5. Calculate sum as number of calls * 10 sec.

The maximum value in seconds which a connection can hold in the 900 sec. interval is calculated as $\text{capacity} * 900\text{sec}$. The connection load is calculated as the sum of durations for all call types divided by the maximum value in seconds. The result is shown as a percentage value (multiplied by 100).

ASR calculations

Assumption: Connection load less than 5% is not representative for ASR calculating.

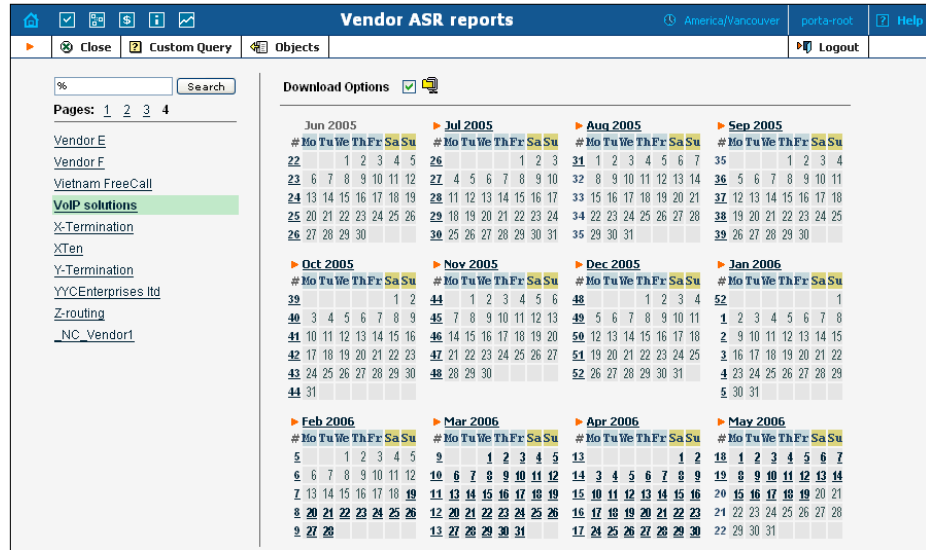
ASR is calculated as the number of connected calls divided by the total number of calls. The result is shown as a percentage value (multiplied by 100).

Relative setup time calculations

Relative setup time is calculated as the ratio between the total setup time of all calls during the given period and the maximum value in seconds ($\text{capacity} * 900\text{sec}$).

ASR Statistics

Follow the **ASR** link from the **Statistics** section in the main menu.



You can download pre-calculated statistics in .CSV format (for previous days) or obtain data online using **Custom Query**. For pre-calculated statistics, click on the vendor at left, then choose the statistics period from the calendar. Pre-calculated statistics will look like the following:

	A	B	C	D	E	F	G	H	I
	Destination	Country	Description	Billable Calls	Calls	ASR, %	Duration, min:sec	ALOC, min:sec	
1	Destination	Country	Description	Billable Calls	Calls	ASR, %	Duration, min:sec	ALOC, min:sec	
2	1		North America	587	264	44.97445	1449:49	5:29	
3	SUBTOTAL			587	264	44.97445	1449:49	5:29	
4	420	CZECH REPUBLIC	Proper	590	297	50.33898	1722:49	5:48	
5	SUBTOTAL			590	297	50.33898	1722:49	5:48	
6	380	UKRAINE	Proper	1171	626	53.45858	3554:22	5:40	
7	SUBTOTAL			1171	626	53.45858	3554:22	5:40	
8	TOTAL			2348	1187	50.55366	6727:0	5:40	
9									
10									

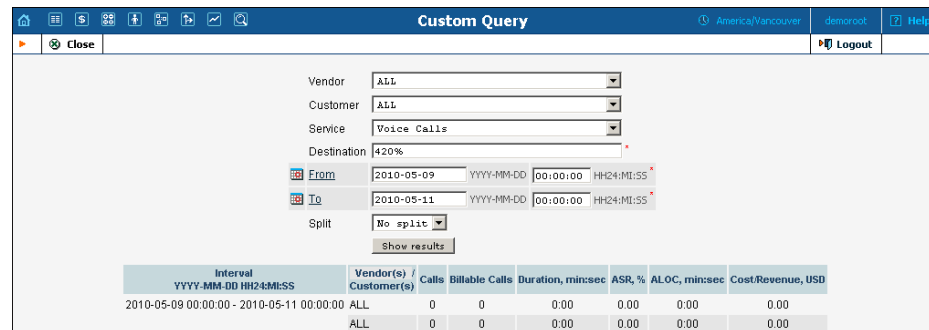
The following data is available:

- Total number of calls
- Number of calls with non-zero duration (billable calls)
- ASR
- Total call duration
- Average Length of Call (ALOC)

Numbers are aggregated per destination prefix, with a subtotal per country and a total for all calls.

Should you need statistics for today, or a report with certain other parameters (for example, divided by hour, so that you see how the ASR

evolved during the day), you can use the **Custom Query** report. This extracts data directly from the database, so use it with caution in order not to overload the server.



Note that both ASR and Cost/Revenue parameters are shown on the **Custom Query** report.

Billing Statistics

Customer xDRs

Lists of all customers' xDRs are calculated daily, and are split into files according to each customer's billing period. These pre-calculated statistics can be automatically mailed to the customer. They are also available for download by the customer on the self-care pages, and for your staff on the admin interface.

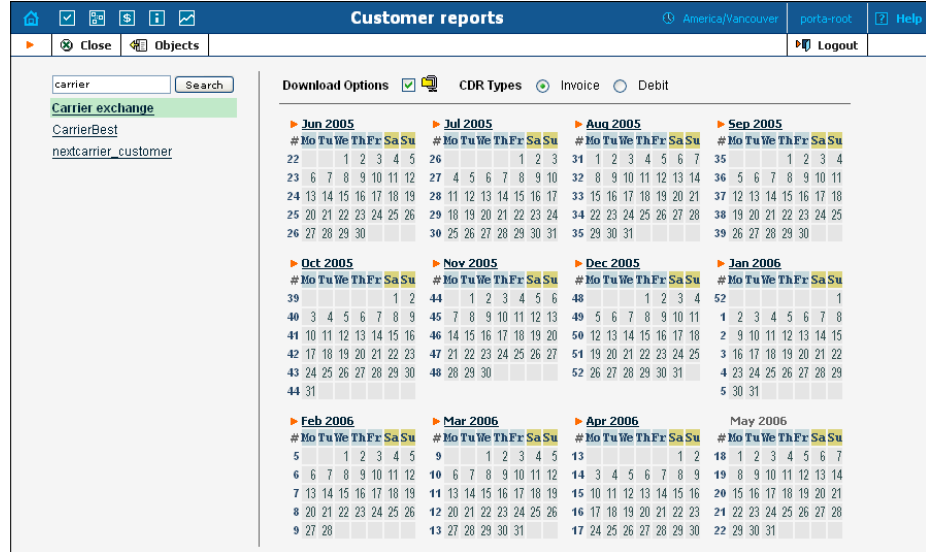
Different types of xDR files are available, depending on the type of customer:

xDR files for retail customers

- **Invoice** – All activities (calls, data access, messaging, etc.) on credit accounts of this customer. These transactions will be included on the customer's invoice for the corresponding billing period.
- **Debit** – Transactions related to activities on debit accounts of this customer. Since debit accounts are prepaid, charges applied to them do not affect the customer's balance or invoice. Therefore, they are included in a separate file so that the customer can easily monitor activities on his debit accounts.

xDR files for resellers

xDRs calculated using the reseller's wholesale tariff. These are the charges applied to the reseller and reflected on his invoice.



Choose the customer and then click on the calendar to obtain statistics for desired period. The .CSV file will have the following structure:

Account	From	To	Country	Description	Connect Time	Charged Time, min:sec	Charged Time, sec.	Charged Amount, USD
56.78.90.1	71866073902	380449313591	UKRAINE	Kiev Region	30-Apr-06 23:59:44	04:24	264	0.616
56.78.90.1	19190767456	420696017957	CZECH REPUBLIC	Proper	30-Apr-06 23:59:05	07:03	423	1.7625
56.78.90.3	60572324893	16049576339	CANADA	British Columbia	30-Apr-06 23:57:14	02:12	132	0.308
56.78.90.1	46890062001	380693412335	UKRAINE	Proper	30-Apr-06 23:56:50	03:52	232	0.58
200.45.23.1	46345311293	380442924858	UKRAINE	Kiev Region	30-Apr-06 23:56:26	02:48	168	0.392
56.78.90.1	82226061971	14257891107	UNITED STATES	Washington	30-Apr-06 23:55:52	02:32	152	0.076
56.78.90.1	57313248607	16047660320	CANADA	British Columbia	30-Apr-06 23:55:33	03:20	200	0.1
56.78.90.1	91438981472	420461329009	CZECH REPUBLIC	Proper	30-Apr-06 23:55:04	03:47	227	0.94584
200.45.23.1	31773453219	420971480263	CZECH REPUBLIC	Proper	30-Apr-06 23:54:41	03:11	191	0.79584
56.78.90.1	68027102122	380975904496	UKRAINE	Proper	30-Apr-06 23:53:42	08:49	529	1.3225
56.78.90.3	28478507435	420802725520	CZECH REPUBLIC	Proper	30-Apr-06 23:44:12	01:17	77	0.32084
56.78.90.1	15383396548	16042029917	CANADA	British Columbia	30-Apr-06 23:44:07	09:54	594	0.297

The following data is available:

- Account ID (if applicable; not available for a reseller’s xDRs, but available for xDRs of a reseller’s sub-customers)
- CLI or ANI (From)
- CLD or DNIS (To)
- Country and description of destination
- Call start time
- Charged time (in minutes:seconds)

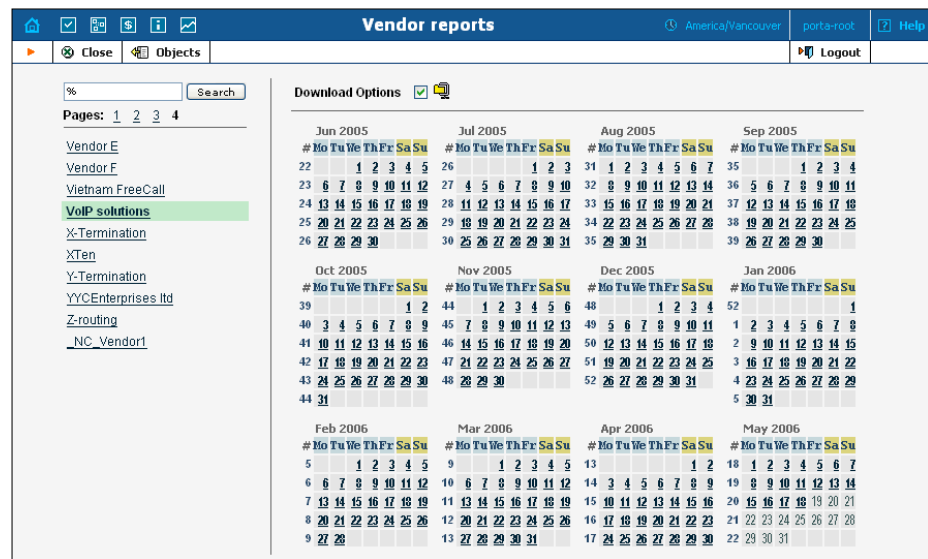
NOTE: When browsing data in Microsoft Excel, extra trailing zeros might be added, for example **345:37** (three hundred forty-five minutes and thirty-seven seconds) will be shown as **345:37:00**. Use proper call formatting in Excel to eliminate this problem.

- Charged time (in seconds)
- Charged amount

NOTE: The call duration shown in the file is based on the charged duration of the call, not the actual call duration. So if, for instance, you bill by 6-second intervals, the total call duration will be higher than the actual duration of the calls.

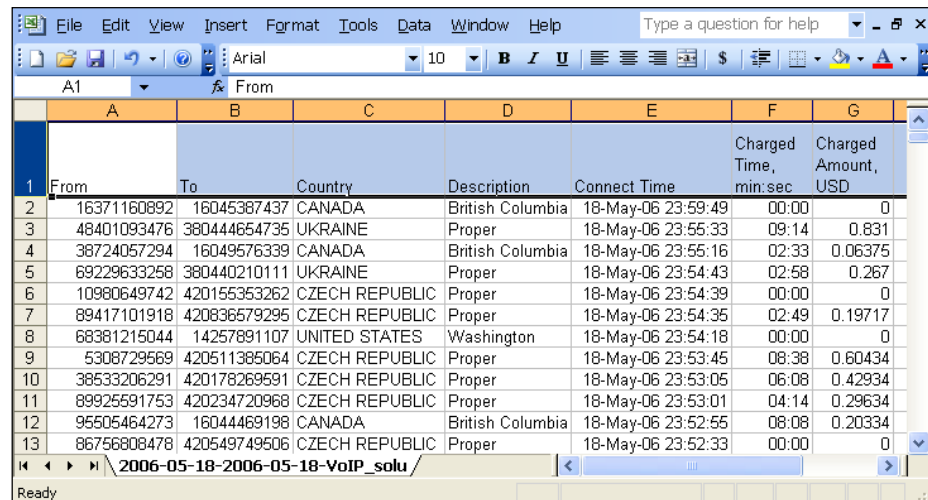
Vendor xDRs

It is very useful to have a detailed list of calls for a specific vendor, in case of disputes over the amount of terminated traffic. These statistics are calculated daily, and are split into files according to the vendor's billing period.



The screenshot shows the 'Vendor reports' interface. On the left, there is a search bar and a list of vendors: Vendor E, Vendor F, Vietnam FreeCall, **VoIP solutions** (highlighted), X-Termination, XTen, Y-Termination, YYEnterprises Ltd, Z-routing, and _NC_VendorI. The main area displays a calendar grid for months from June 2005 to May 2006. Each month has a small table with days of the week (Mo, Tu, We, Th, Fr, Sa, Su) and numbers. A 'Download Options' checkbox is checked. The interface also includes navigation buttons like 'Close', 'Objects', and 'Logout'.

Choose the vendor and then click on the calendar to download statistics for the desired period. The .CSV file will have the following structure:



	A	B	C	D	E	F	G
1	From	To	Country	Description	Connect Time	Charged Time, min:sec	Charged Amount, USD
2	16371160892	16045387437	CANADA	British Columbia	18-May-06 23:59:49	00:00	0
3	48401093476	380444654735	UKRAINE	Proper	18-May-06 23:55:33	09:14	0.831
4	38724057294	16049576339	CANADA	British Columbia	18-May-06 23:55:16	02:33	0.06375
5	69229633258	380440210111	UKRAINE	Proper	18-May-06 23:54:43	02:58	0.267
6	109806489742	420155353262	CZECH REPUBLIC	Proper	18-May-06 23:54:39	00:00	0
7	89417101918	420836579295	CZECH REPUBLIC	Proper	18-May-06 23:54:35	02:49	0.19717
8	68381215044	14257891107	UNITED STATES	Washington	18-May-06 23:54:18	00:00	0
9	5308729569	420511385064	CZECH REPUBLIC	Proper	18-May-06 23:53:45	08:38	0.60434
10	38533206291	420178269591	CZECH REPUBLIC	Proper	18-May-06 23:53:05	06:08	0.42934
11	89925591753	420234720968	CZECH REPUBLIC	Proper	18-May-06 23:53:01	04:14	0.29634
12	95505464273	16044469198	CANADA	British Columbia	18-May-06 23:52:55	08:08	0.20334
13	86756808478	420549749506	CZECH REPUBLIC	Proper	18-May-06 23:52:33	00:00	0

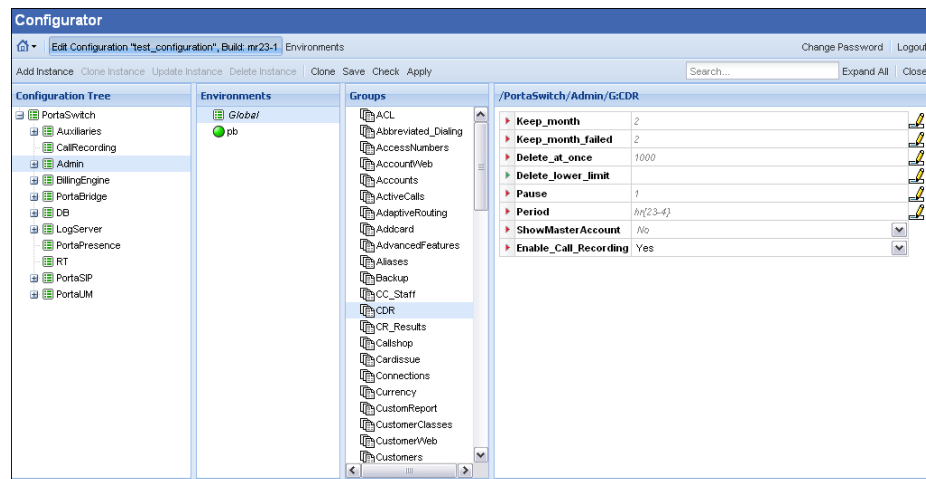
The following data is available:

- CLI or ANI (From)
- CLD or DNIS (To)
- Country and description of destination
- Call start time
- Charged time
- Charged amount

xDR Cleanup Procedure

xDRs (do not mix with billing statistics) are stored in the Master database and replicated to the Slave database. In order to prevent system overgrowing, xDRs are periodically cleaned up by performing a special task. To correctly reflect a charge history a special “Cleanup” xDR exists that accumulates the charged amounts and times from all the xDRs that have been cleaned up for every account / customer / vendor. This xDR shows up as a “Cleaned up transaction” on the xDR history page of the web interface.

By default, the Cleanup procedure occurs during each off-peak period (every night) for all the xDRs that are more than 2 months old, though these options are configurable to meet special requirements via the Configurator.



/PortaSwitch/Admin/GCDR	
Keep_month	2
Keep_month_failed	2
Delete_at_once	1000
Delete_lower_limit	
Pause	1
Period	h\23-4}
ShowMasterAccount	No
Enable_Call_Recording	Yes



Although all the statistics / invoices are calculated up to the moment of the specified xDRs' Cleanup, you can increase the period for keeping old xDRs according to your needs.

Cost/Revenue Statistics

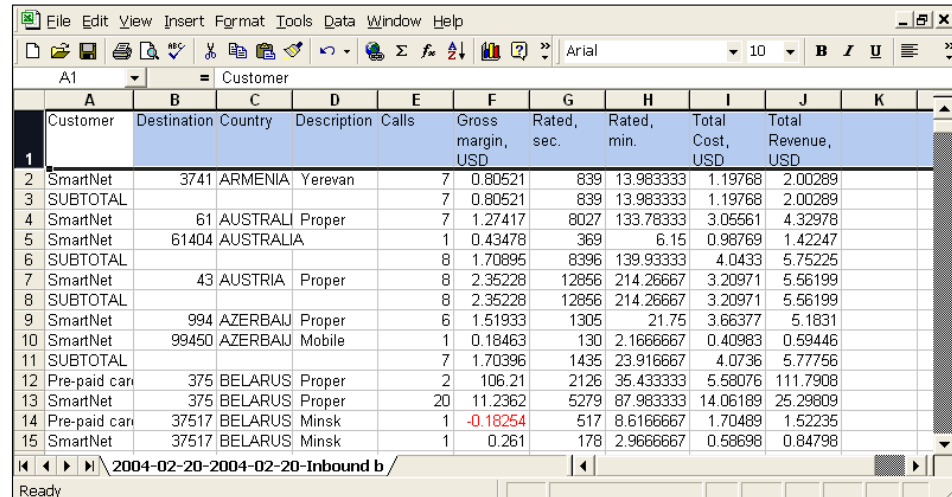
These are essential tools for monitoring the growth of your business. Working with different partners, different currencies and different prices, it is very easy to make mistakes and carry out non-profitable calls.

Cost/Revenue reports allow you to monitor this and stay out of trouble. To access Cost/Revenue reports, follow the **Cost/Revenue reports** link from the **Statistics** section of the main menu.



There are five variants of pre-calculated reports:

- By customer and destination, subtotal per country
- By customer and destination, subtotal per customer
- By vendor and destination, subtotal per country
- By vendor and destination, subtotal per vendor
- By destination



Customer	Destination	Country	Description	Calls	Gross margin, USD	Rated, sec.	Rated, min.	Total Cost, USD	Total Revenue, USD
SmartNet	3741	ARMENIA	Yerevan	7	0.80521	839	13.983333	1.19768	2.00289
SUBTOTAL				7	0.80521	839	13.983333	1.19768	2.00289
SmartNet	61	AUSTRALI	Proper	7	1.27417	8027	133.783333	3.05561	4.32978
SmartNet	61404	AUSTRALIA		1	0.43478	369	6.15	0.98769	1.42247
SUBTOTAL				8	1.70895	8396	139.933333	4.0433	5.75225
SmartNet	43	AUSTRIA	Proper	8	2.35228	12856	214.26667	3.20971	5.56199
SUBTOTAL				8	2.35228	12856	214.26667	3.20971	5.56199
SmartNet	994	AZERBAJ	Proper	6	1.51933	1305	21.75	3.66377	5.1831
SmartNet	99450	AZERBAJ	Mobile	1	0.18463	130	2.1666667	0.40983	0.59446
SUBTOTAL				7	1.70396	1435	23.916667	4.0736	5.77756
Pre-paid can	375	BELARUS	Proper	2	106.21	2126	35.433333	5.58076	111.7908
SmartNet	375	BELARUS	Proper	20	11.2362	5279	87.983333	14.06189	25.29809
Pre-paid can	37517	BELARUS	Minsk	1	-0.18254	517	8.6166667	1.70489	1.52235
SmartNet	37517	BELARUS	Minsk	1	0.261	178	2.9666667	0.58698	0.84798

The following data is available:

- Name of customer/vendor
- Destination prefix
- Country and description of destination
- Total number of successful calls

- Gross margin – The difference between total revenue and the cost for this destination
- Rated, sec. – The sum of the time charged for all calls to this destination in seconds
- Rated, min. – The sum of the time charged for all calls to this destination in minutes. This is the same value as in the previous column, only expressed in different units. Thus, for instance, if *Rated, sec* contains **180**, here **3.0** will be shown.
- Total Cost – Summary of the amount charged for all the vendor's CDRs for this destination. If the vendor uses a currency different than your base one, this will be converted using the current exchange rate.
- Total Revenue – Summary of the amount charged for all the customer's CDRs for this destination. If the customer uses a currency different than your base one, this will be converted using the current exchange rate. Note that, in the case of resellers, we use the value of the reseller's CDR, not the sub-customer's, since your revenue is what gets invoiced to the reseller.

In addition, you can use the **Custom Query** report, which is identical to the **Custom Query** report in the ASR statistics.

8. How to ...


Log in to PortaBilling® after installation

The default username is “pb-root” and the password is “pb-root” as well. After you login for the first time, you will be requested to change the password – please remember the new password you choose.

Locate the h323-conf-id for a call

Unique call IDs are extremely important for troubleshooting. They allow you locate a call in the database and find call information in the billing engine logs or RADIUS detail files. Finally (and this is the most important thing), when reporting call problems to the PortaOne Support team or your business partner, a call ID helps to prevent confusion and allows the problem to be solved quickly.

In order to find the h323-conf-id for a call, do the following:

1. From the main menu, go to **Trace Session**.
2. Using the CLD and from/to date filters, make sure the required call is shown in the list of calls on the screen.
3. In the row containing call information, click on the  **View** icon (far left column). You will be shown detailed information about the call.
4. The first field on the screen should be the h323-conf-id, which is a combination of four hexadecimal numbers separated by spaces, e.g. 5FF7F6D1 715E02C6 A40990F3 C823E27E.

Troubleshoot an incorrectly billed call

1. Make sure that someone in your organization is subscribed to the PortaBilling® mail alerts (especially “Missing critical information”). This will help you to detect problems early.
2. Find the h323-conf-id for this call. This is a unique ID (a string of four hex numbers) generated by the gateway when the call was started, which will help you to exactly identify this particular call among all the others. You can find this ID by doing one of the following:
 - Looking at the statistics on your gateway.
 - In the **BE Log Viewer** section of the web interface you can find a list of recent calls (a select menu allows you to see all calls made in the past 5, 10, 30, etc. minutes).
3. Use **BE Log Viewer** on the web interface to find all of the information about call processing there.

Note: Normally we receive information about each of the call legs separately, so it is necessary that you check the log entry regarding the processing of **all** call legs and the final call clean-up.

4. The following are typical error situations:
 - **Call was not billed at all.** It was considered an “unresolved” call, because we did not detect that it went to any of the vendors. Check that you have correctly defined connections to your vendors.
 - **Call was billed only to vendor, not to an account or customer** (and you received an email alert). The billing engine needs an Account ID in the User-Name attribute to correctly identify the account and customer. Check the logs to see what was in the User-Name attribute. Typical situations are:
 1. **Incorrect value in User-Name** (for example, phone number instead of IP address of the remote gateway). Cause: the required application was not used to handle the call. Remedy: check that the application is configured and associated with the corresponding *incoming* dial-peer.
 2. **On some of the call legs there is a correct value in the User-Name, while on others there is an IP address.** When a call is originated on gateway A and terminated on gateway B, then a “real” username appears only in the accounting from gateway A. In gateway’s B accounting the username will be the IP address of gateway A, because gateway A had to authenticate itself before being able to make the call. This is a perfectly normal situation. PortaBilling® recognizes this, and will replace a username identical to the node’s ID (IP address) with a “real” username from the other call legs.
 3. **Value in User-Name is correct, but reports “Did not find account/customer”.** Check the accessibility for the account’s product. It may be that, even if you have such an account, its accessibility is incorrect.
 - **Call was billed, but the phone number in the billing is incorrect (not in E.164 format) and there is a “Mismatch in rates or destinations” error.** Cause: missing or incorrect translation rule on connection to the vendor. Remedy: assign a translation rule; read the [PortaSwitch Wholesale](#)

and [Traffic Exchange Services Handbook](#) (*Handle technical prefixes and numbering formats* section).

- **Call was billed and the phone number in the billing is correct, but there is a “Mismatch in rates or destinations” error.** Cause: missing rate for this phone prefix in the tariff. Remedy: create a rate for this destination in the tariff.
5. If you are unable to solve the problem by yourself, submit a problem report to support@portaone.com. Please make sure that you include the following in your email:
- A detailed description of the call flow and what seems to be incorrect.
 - H323-conf-id of the call.
 - Relevant items from the billing engine log.

Make the Periodic Payments tab appear in the customer/account info

The Periodical payments tab will only be shown if the rest of the system is configured to accept payments. This requires:


- At least one payment system defined in the **Payments** section; this payment system must have the **Recurring** option enabled.
- A payment system associated with the currency of this account or customer.
- For accounts, the **E-commerce enabled** flag must also be switched on.
- On the **Payment info** tab, the Preferred payment method must be set and information about the credit card entered.

For sub-customers or accounts belonging to sub-customers, the requirements are the same, except that the payment systems are defined in the reseller information, not in Payments section.

Make the Custom Fields tab appear in the customer / account info

Custom fields should be initially set up using the **Custom Fields** tab on the **Web Interface** page. Only when configured will the corresponding tab appear in the customer / account info. Using custom fields, you can store a set of extra attributes (e.g. driver's license ID or tax code) that supplements the standard PortaBilling100 information.

For each new custom field, the following attributes must be set:

Field	Description
Object	Defines whether the custom field applies to the Customer or the Account.
Name	The descriptive name of the field. This is the name that will be displayed next to the custom field on the Customer / Account Info page.
Type	Defines the type of field: <ul style="list-style-type: none"> • Text – basic single-line input field; • Number – input field used to store and validate numerical values; • Date – field type used to store dates; • Date & Time – custom field that stores dates with a time component; • List – single select list with a configurable set of options.
Properties	Enables you to customize properties of the field that define its form, appearance, or value. These properties are specific to the field type. Click Properties or the wizard icon  to invoke the wizard. This will enable you to define a new field format or change an existing one and to specify the default value a custom field should have.
Default	Read-only attribute which must be specified in the Properties attribute.
Mandatory	Defines the mandatory status of the field.



You can delete a custom field at any time. All records of its values will then also be deleted.

Make a custom report from PortaBilling®

PortaBilling® provides you with an open data model. An ER-diagram of the database structure is not included in this document, but may be obtained from Porta Software Ltd upon request. If you want to prepare custom reports on your workstation or a non-PortaBilling® server, you will need to do the following:

- Make sure the remote computer has database drivers installed to access the PortaBilling® database. Normally you would use native MySQL connectivity on Unix-based hosts and ODBC on Windows-based hosts.
- For any data-mining solutions (extracting data from the database), use **only** the slave database.

- Use a tool like Crystal Reports, Microsoft Access or some custom application to retrieve data from the database, process it and submit it to the user.

Use ODBC to connect to PortaBilling®

ODBC (Open Database Connectivity) provides a way for client programs to access a wide range of databases or data sources. If you need extended customized reporting not available in PortaBilling®, you can do this using external tools such as MS Access or Crystal Reports.

Create a MySQL user to be used for reports

1. Login into your PortaBilling® slave server using ssh.

2. Start the MySQL command tool.

```
andrew@demo:/home/porta-admin$mysql -u root mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 42122 to server version: 4.0.17-log
Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.
mysql>
```

3. Create a new user using the GRANT command.

```
mysql> grant ALL PRIVILEGES on `porta-billing`.* to
`reports`@'192.168.0.5' identified by 'pod23uk';
Query OK, 0 rows affected (0.02 sec)
```

```
mysql>
```



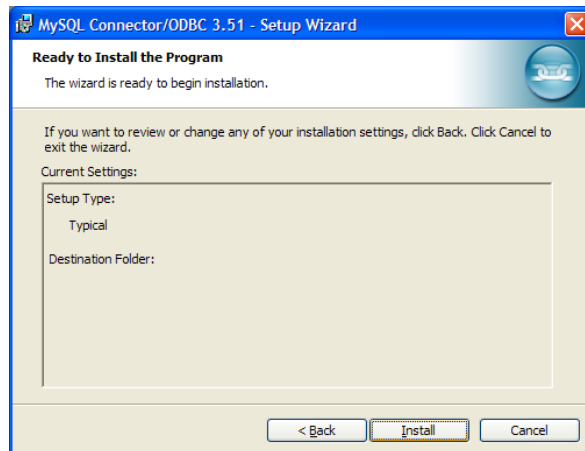
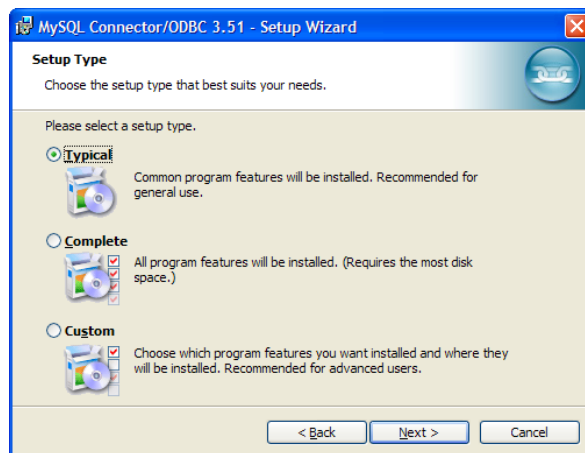
NOTE: The command above will permit access to all of the tables in the database. It is provided only as an example. You can modify it according to your actual needs, e.g. give **read-only** access to **some** tables only.

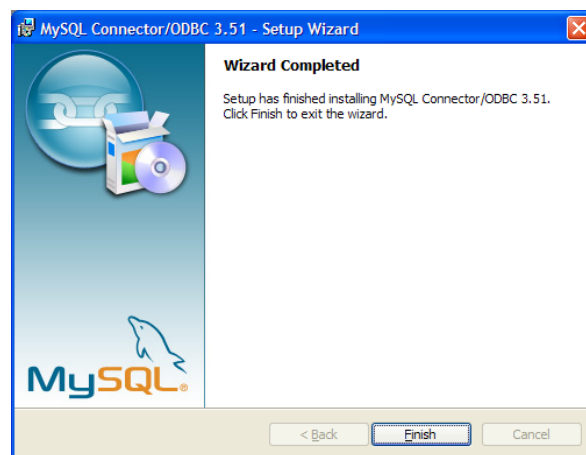
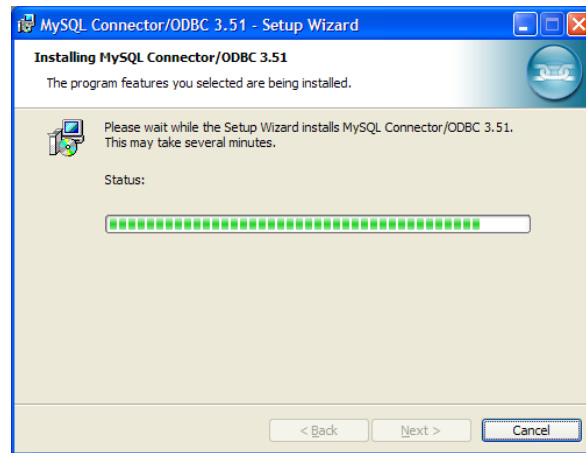
4. Flush the privileges.

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.07 sec)
```

Installing the MySQL ODBC driver

1. Download and run the installation package from:
<http://www.mysql.com>
2. Click **Next** on the information screens.





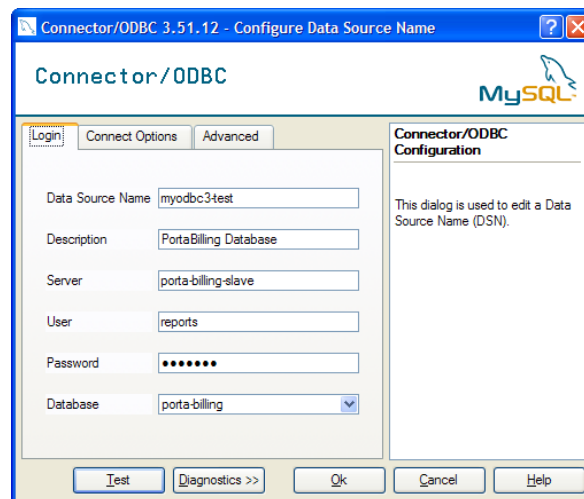
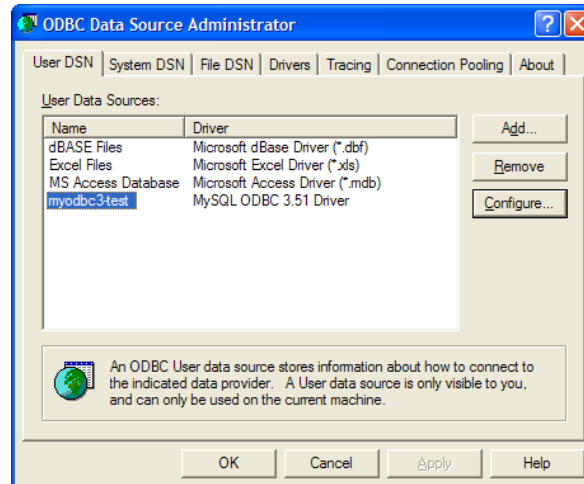
3. Click **Finish**.

Configuring ODBC

Before configuring the data source, create an MySQL user on slave DB with read-only permissions. Please examine the following document on how to add new user accounts to MySQL:

http://www.mysql.com/doc/en/Adding_users.html

1. Control panel -> Administrative tools -> Data sources (ODBC).
2. Select myodbc3-test and click **Configure...**. Fill in the configuration form.

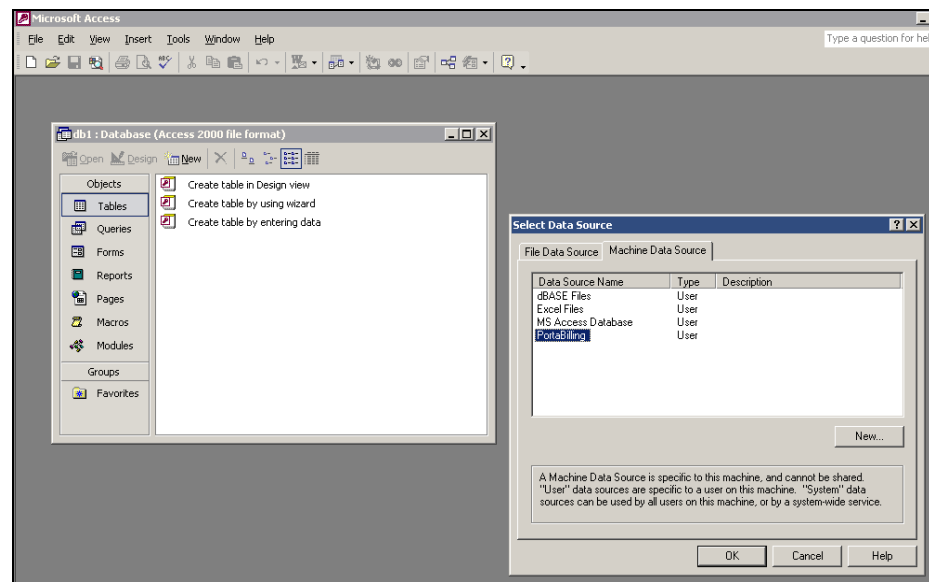


Important parameters include:

- **Host/Server Name (or IP)** – hostname (or IP address) of your slave server
- **Database Name** – porta-billing
- **User, Password** – username and password of the MySQL user you have created for reporting purposes
- **Port** – the port on which the database service is accessible; enter **3307** here

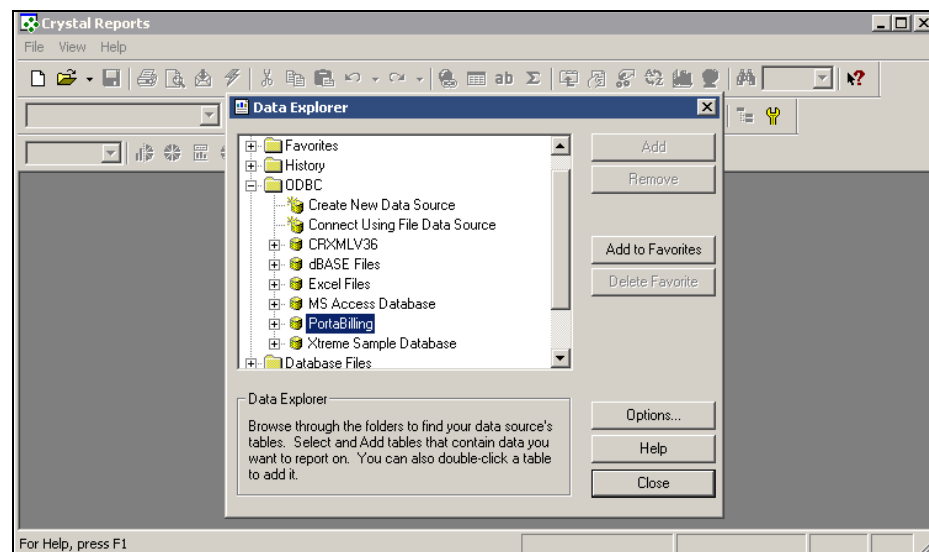
Note: This port number differs from the one used by default.

Using ODBC



In MS Access:

1. Create a blank database.
2. Right-click in the table design view, and choose **Link tables...**
3. Choose **ODBC databases** from “Files of type...” list.
4. Select **Machine data source**.
5. Select **PortaBilling** and click OK.
6. Select the desired tables.



In Crystal Reports:

1. Create a *New Report*.
2. In *Data Explorer*, open **ODBC** branch.
3. Select **PortaBilling**.
4. Select the desired tables.

Force PortaBilling® to disconnect after a customer calls over his credit limit

There is no need for PortaBilling® to do this, as the gateway is able to by itself. When the gateway authorizes an account to make a call in PortaBilling®, PortaBilling® returns a maximum credit time in the case of a successful authorization. When the gateway connects the call, it starts a timer; when the timer hits zero, it automatically disconnects the call. PortaBilling® sends **two** RADIUS attributes related to the maximum call duration. The `h323-credit-time` attribute is the standard one (announced credit time); there is also a custom `h323-ivr-in = DURATION` attribute (actual credit time). The value for the second attribute is computed with billing tricks taken into account. Thus, in the case of \$10 available funds, a \$0.10/min rate and a 10% post-call surcharge, this attribute will be 5400 seconds (90 minutes), while the `h323-credit-time` will be 6000 seconds (100 minutes).

If the application supports only the `h323-credit-time` attribute (e.g. Cisco default debit card application), then this time will be announced to the end user and used to disconnect the call, and `h323-ivr-in = DURATION` will be ignored.

Create accounts to be used for SIP services

There are no special requirements as to how such accounts should be created. You use the same interface to create and manage accounts for all services supported by PortaBilling® (H323, SIP). Thereafter accounts can use H323, SIP or SIP & H323 services, depending on their product's accessibility. So if you plan for accounts with a certain product to be able to login to the SIP server and make outgoing calls, be sure to include the PortaSIP® node with the appropriate tariff in the **Accessibility** for this product.

Integrate PB logins to your website

You can include the login form on your website using the following HTML code:

```
<form name="log" method="post"
action="https://pb.mycompany.com/index.html">
  Login <input type="text" name="pb_auth_user" value="">
  Password <input type="password" name="pb_auth_password"
value="">
  <input type="submit" value="Login">
```

```
<input type="hidden" name="redirectOnLoginError"
value="http://www.myweb.com">
</form>
```

"redirectOnLoginError" - in case of unsuccessful login, PortaBilling® will redirect the browser to the URL specified in the value attribute.

Add your company name and logo to every PortaBilling® web page

1. The custom HTML component file should be placed in the directory accessible by the web server; the best place is in the `~porta-admin/apache` directory. Login to the slave server and become a super-user, then type:

```
cd ~porta-admin/apache
mkdir brandpane
```
2. Create the HTML component which will be included on every PortaBilling® page. Edit the file `my.html` in the `~porta-admin/apache/brandpane` directory (a sample file is shown below).

```
<table width=100% border="0" align="left"
cellpadding="0" cellspacing="0" bgcolor="white">
  <tr>
    <td width="100%">
      
    </td>
  </tr>
</table>
```

3. Activate the brandpane feature in the PortaBilling® web interface config (edit the `~porta-admin/etc/porta-admin.conf` file on the slave server).

```
[Web]
...
BrandPane=/brandpane/my.html
...
```

If you are not familiar with editing text files on the Unix platform, ask PortaOne support to do this for you.
4. Restart the Apache web server (`apachectl restart`). Now you should see your company logo at the top of every PortaBilling® web page.

Add a reseller's logo to his customers' web pages

1. Activate the brandpane feature as described in the topic above.

2. Configure an additional virtual web server which will run on the reseller's domain name (e.g. www.supercall.com).
3. Include conditional statements in the brandpane component. These will analyze the domain name the customer is trying to access, and will then produce the company name and logo for your domain (www.sipcall.com), while for the reseller's (www.supercall.com) the reseller's name and logo will be produced. See the following example:

```
% if ($ENV{HTTP_HOST} =~ m/supercall.com/) {  
  
% } else { # our default domain  
  
% }
```


Frequently Asked Questions (FAQ)

What do I need the xDR Cleanup procedure for? What would be the impact on my system if I disable the Cleanup?

Database size continually increases, thus affecting system performance: longer PDD (Post Dial Delay), longer response time on WI pages related to xDRs (rate upload, xDR browser, invoicing) etc. Large xDRs tables also increase database query execution time. By performing the xDR Cleanup procedure your system continues working smoothly.

What period should I choose for keeping xDRs?

It depends on your requirements. We recommend keeping xDRs for the period that corresponds to the maximum period you need to generate reports plus an additional month. Thus, if you only need monthly reports, keeping the xDRs for 2 months will be sufficient.

What time period should I choose for xDR Cleanup?

The Cleanup procedure may cause an additional load on both the Master and Slave databases. For this reason, we recommend it to be executed only during off-peak hours.

Glossary / List of Abbreviations

A

AAA – Authentication, Authorization and Accounting. A suite of network security services that provides the primary framework through which you can set up access control on your router or access server.

ACD – Automated Call Distribution

AD – Abbreviated Dialing

ADSL – Asymmetric Digital Subscriber Line

AIS – Automated Information System

ANI – Automatic Number Identification

API – Application Programming Interface

ART – Audible Ringing Tone

AS – Answer Supervision

ASP – Application Service Provider

ASR – Answer Seizure Ratio

ATA – Analog Telephone Adapter

AVVID – Architecture for Voice, Video and Integrated Data

B

B2B – Business to Business

B2C – Business to Consumer

BH – Busy Hour

BHCA – Busy Hour Call Attempts

BHCC – Busy Hour Call Completion

BIOS – Basic Input/Output System

bps – Bits per Second

Bps – Bytes per Second

C

call leg – A link or hop along the route from the calling party to the destination (called) party.

CALEA – Communications Assistance for Law Enforcement Act

CDR – Call Detail Record

CGI – Common Gateway Interface

CHAP – Challenge Handshake Authentication Protocol

CLE – Customer Located Equipment

CLEC – Competitive Local Exchange Carrier

CLI – Command Line Interface

CLI, CLID – Calling Line Identification

CMS – Call Management Services

CPE – Customer Premise Equipment

CRM – Customer Relationship Management

D

DCE – Data Communications Equipment
DHCP – Dynamic Host Configuration Protocol
DID – Direct Inward Dialing
DMZ – Demilitarized Zone
DNS – Domain Name System
DSL – Digital Subscriber Line
DTE – Data Terminal Equipment
DTMF – Dual-tone multi-frequency. The use of two simultaneous voice-band tones for dialing (such as touch-tone).

E

E-1 – Europa-1, the European equivalent to T1 (but higher bandwidth).
E911 – Enhanced 911
ENUM – Electronic Numbering

F

FAQ – Frequently Asked Questions
FCC – Federal Communications Commission
FoIP – Fax over IP
FQDN – Fully qualified domain name. The full name of a system, rather than just its hostname. For example, test is a hostname, while test.portaone.com is an FQDN.
FXO – Foreign Exchange Office
FXS – Foreign Exchange Station

G

GK – Gatekeeper. The entity that maintains a registry of devices in an h323 network.
GMT – Greenwich Mean Time
GSM – Global System for Mobile Communications
GUI – Graphical User Interface
GUID – Globally Unique Identifier (also called h323-conf-id). The identifier for end-to-end calls. A new GUID is assigned to each new call. There may be numerous calls within a single session.

H

HDD – Hard Disk Drive
HTML – Hypertext Markup Language
HTTP – Hypertext Transfer Protocol

I

ICMP – Internet Control Message Protocol
IM – Instant Messaging

IMAP – Internet Message Access Protocol
IMS – IP Multimedia Subsystem
IOS – Internetworking Operating System
IP PBX – Internet Protocol Private Branch Exchange
ISDN – Integrated Services Digital Network
ISP – Internet Service Provider
ISV – Independent Software Vendor
ITSP – Internet Telephony Service Provider
IVM – Instant Voice Messaging
IVR – Interactive Voice Response
IXC – Inter-Exchange Carrier

J

JMS – Java Messaging Service
JVM – Java Virtual Machine

L

LAN – Local Area Network
LBS – Location Based Services
LCR – Least Cost Routing
LDAP – Lightweight Directory Access Protocol
LEC – Local Exchange Carrier
Long pound calls – see *Multi-session call*

M

MAC – Media Access Control
MAC address – The physical address of a device on Ethernet.
MG – Media Gateway
MGCP – Media Gateway Control Protocol
MIME – Multi-Purpose Internet Mail Extension
MMS – Multimedia Messaging Service
MOD – Movies on Demand
Multi-session call – A call in which several outgoing calls to different destinations can be made under the same incoming call to the IVR.

N

NANP – North American Numbering Plan
NAS – Network Access Server
NAT – Network Address Translation
NBH – Network Busy Hour
NGN – Next Generation Network
NIC – Network Interface Card
NNX – The current general configuration for exchange codes within each area code.
NOC – Network Operations Center

NPA – Numbering Plan Area
NTP – Network Time Protocol

O

OEM – Original Equipment Manufacturer
OLTP – Online Transaction Processing
OS – Operating System

P

PABX – Private Automatic Branch Exchange
PAP – Password Authentication Protocol
PBX – Private Branch Exchange
PDF – Portable Document Format
PIN – Personal Identification Number
PnP – Plug and Play
POP – Point of Presence
POTS – Plain Old Telephony Service. Basic telephone service supplying standard single-line telephones, telephone lines, and access to PSTN.
PPP – Point-to-Point Protocol
PRI – Primary Rate Interface
PSAP – Public Service Answering Point
PSTN – Public Switched Telephone Network

Q

QoS – Quality of Service

R

RADIUS – Remote Authentication Dial-In User Service
RAM – Random Access Memory
RAS – Registration, Admission and Status protocol
RDBMS – Relational Data Base Management System
RFP – Request for Proposal
RFQ – Request for Quote
RJ – Registered Jack
RS-232 – Recommended Standard 232 (physical interface between DCE and DTE)
RTP – Real-Time Transfer Protocol
RTCP – Real-Time Control Protocol
RTSP – Real-Time Streaming Protocol

S

SBC – Session Border Controller
SCSI – Small Computer Systems Interface
SDK – Software Developers Kit
SDP – Session Definition/Description Protocol

SG – Signaling Gateway
SIGTRAN – Signaling Transport
SIM – Subscriber Identify Module
SIM Card – Subscriber Identify Module Card
SIP – Session Initiation Protocol
SIP-T – Session Initiation Protocol for Telephony
SLA – Service Level Agreement
SMP – Symmetric Multiprocessing
SMS – Short Messaging Service
SMTP – Simple Mail Transfer Protocol
SOAP – Simple Object Access Protocol
SOHO – Small Office/Home Office
SQL – Structured Query Language
SS7 – Signal System 7

T

T.37 – Standard for store-and-forward FoIP (Fax over IP).
T.38 – Standard for real-time FoIP (Fax over IP).
TAPI – Telephony Application Program Interface
TCL – Tool Command Language. A scripting language used for gateway products both internal and external to the Cisco IOS software code.
TCP – Transmission Control Protocol
TCP/IP – Transmission Control Protocol/Internet Protocol
Telco – A telephone company, such as LEC.
TFTP – Trivial File Transfer Protocol
TTS – Text-to-Speech
TVoIP – TV over IP

U

UA – User Agent
URI – Uniform Resource Identifier
URL – Uniform Resource Locator
USB – Universal Serial Bus

V

VAD – Voice Activity Detection
VAR – Value-Added Reseller
VLAN – Virtual LAN
VOD – Video on Demand
VoiceXML – Voice Extensible Markup Language
VoIP – Voice over Internet Protocol

W

WAN – Wide Area Network
WiFi – Wireless Fidelity

WPA – WiFi Protected Access

WWW – World Wide Web

X,Y,Z

xDR – Extensible Detail Record

XML – Extensible Markup Language