PORTA
ONE

Porta Billing100™

Templates Guide

Maintenance Release 12

Copyright notice & disclaimers

Copyright © 2000-2006 PortaOne, Inc. All rights reserved

PortaBilling Templates Guide, July 2006
Maintenance Release 12
V1.12.1

Table of contents

Preface

The PortaBilling100 Templates Guide gives a detailed explanation of the template module. This manual also describes in detail how to optimally import and export data from PortaBilling.

Where to get the latest version of this guide

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

Conventions

This publication uses the following conventions:
- Commands and keywords are given in **boldface**
- Terminal sessions, console screens, or system file names are displayed in `fixed width font`

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

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

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

Tips provide information that might help you solve a problem.

1. Template Concepts

Real-life situation

Before discussing some technical details about PortaBilling and templates, let's take a look at a "real life" example which will help us learn more.

Situation

You are the owner of a VoIP billing system. You receive list of rates from your vendors, which you need to import into the system.

Solution 1: Manual entry

In order to do this, you decide to hire a temporary worker to type in all the data. This person does not know much about your business, so you will have to give him or her detailed instructions on how to proceed. These might be similar to the following:
1. This is an Excel file from our vendor.
2. After you open it, in the first row of the second worksheet, entitled "Rates", you will see a title and a list of rates underneath it.
3. The value in column C (entitled "Prefix") goes in the "Destination" field on the web interface, but do not enter **011** at the beginning of the number. The value in column F (entitled "Price per minute") goes in the "Price 1" and "Price N" fields on the web. There is no "Effective From" data in the file, so just enter the current date in that field on the web.

Does this sound complicated? Let's now take a look at what you have just done:
- You have specified the file format which the data is in ("**an Excel file**").
- You have specified where the required data is to be found within that file ("**in the first row of the second worksheet, entitled 'Rates'**").
- And, finally, for each data element which needs to be imported into the system (Destination, Price) you have specified:
    - Which column the data is in ("**column C (entitled 'Prefix')**").
    - The format of the data (whether it needs to be changed somehow prior to being entered into the system) ("**do not enter 011 at the beginning**").

Now, let's take a look at an alternative way of handling the same process more easily, faster, and with fewer errors.

Solution 2: Template engine

You can use the PortaBilling template engine to handle the import or export of data from PortaBilling. When using templates in PortaBilling, you are in a situation similar to that described above. Since the template processor knows nothing about the format of your data, you will have to provide it with detailed instructions similar to those you gave to the person in our example. You will have to tell the template processor about the file format of the input file, where the useful data is located inside that file, and how to format the data. Of course, you will enter these instructions as a template definition, so that:

- They remain stored in the system and you can always reuse them later.
- Everything is specified explicitly, with no communication errors.
- Since template engine processing is fast and automatic, you do not need to wait for human data entry (which might take days).

Template engine

The template engine allows you to define data conversion rules (templates) and execute them. There are two methods of template-based data processing and, accordingly, two types of templates.

Import data into PortaBilling (upload)
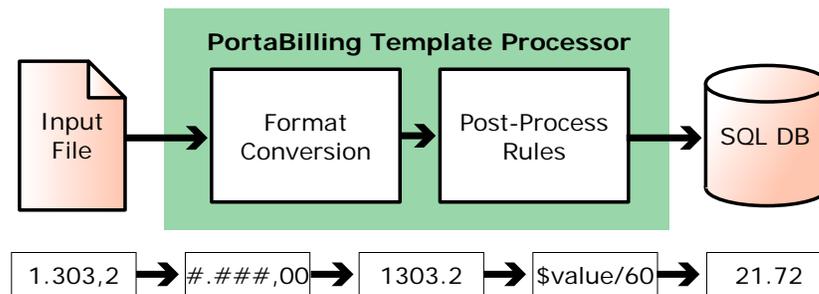
Figure 1-1

Such a template (called an upload template) defines:

- The input file type (for example, "CSV" or "Excel")
- The location of the input file
- The format of the file
- Changes to apply to the data before entering it into the database

When data processing occurs, the template engine parses the input file according to its type, locates the data inside the file, converts it into an internal representation according to the format specification, applies post-processing rules, and then writes the data to the database.

Export data from PortaBilling (download)
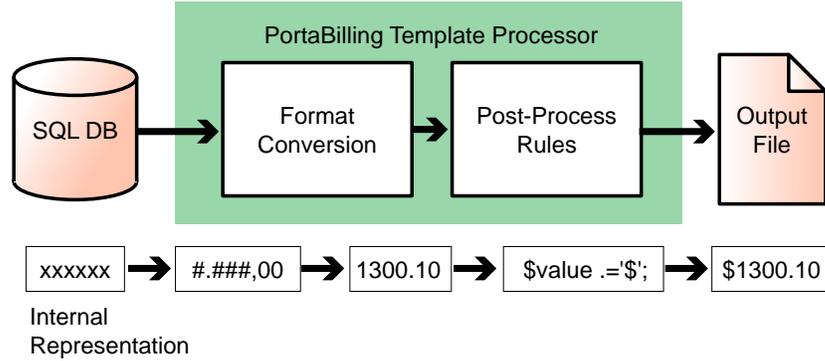
Figure 1-2

Such a template (called a download template) defines:

- Which format the data should be represented in
- How the data is to be processed before being entered into the database
- How the data should be arranged in the output file
- The output data format

When data processing takes place, the template engine extracts data from the database, converts it into the specified format, applies post-processing rules, and then creates a file of the specified type, entering the data in the appropriate places in the file.

Template components

Data for an invoice or a tariff may have a very complex structure. In order to facilitate such cases, data within the template is split into different groups, called "template components", so that it can be managed separately.

Look at the sample invoice in the picture below:

Page 1 of 2

Porta Software Ltd
902 - 2181 West 38th Ave.
Vancouver
V6M 1R8

Your company logo

Invoice

Date	Invoice #
2004-03-30	22

Bill To:

Retail test Customer
Mr. Vasiliy N Ivanov
Wall str., 22
New-York
USA

Statement Period			
From	To	Terms	Due Date
2004-02-11	2004-02-11	Due on receipt	'Invoices.due_date' not

Description	Amount
Voice Over IP Services	12.17 USD
GST 7%	0.85 USD
PST 7.5%	0.91 USD
Previous balance	0.00 USD

Figure 1-3

We can clearly identify a few independent groups of data here:

1. **Invoice header** – This is present only once in the file, and contains global information.
2. **Invoice line(s)** – Groups of data which repeat multiple times in the file, each time with different values.
3. **Invoice footer** – This is present only once in the file, and contains global information.

Here is another example, this time for a tariff:

Figure 1-4

Again, we have several different groups here:

1. **Header** – This is present only once in the file, and contains global information.
2. **Tariff parameters** – This is also present only once in the file, and likewise contains global information.
3. **Rate data** – A data grouping which repeats multiple times in the file, with different values.

Thus there are six possible template components:

Header

A header contains data fields with global data for the object. On an invoice, for example, this would be the invoice number or the name of the company issuing the invoice. A tariff header contains the tariff name, currency and description.
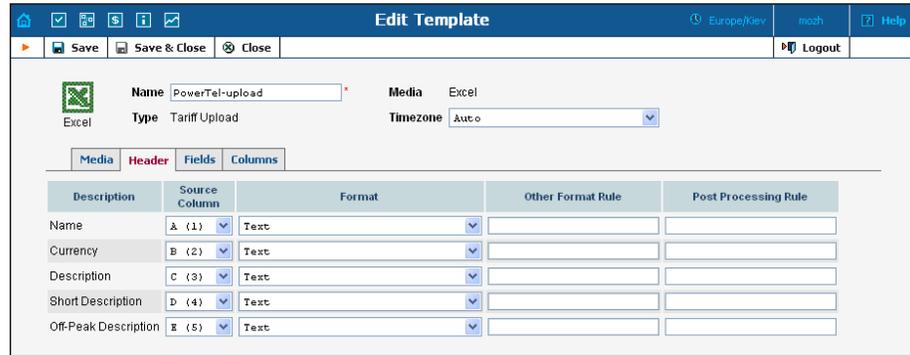
Figure 1-5

Page Header

This template component is available only for printable forms in which data is split across several pages. (For CSV or Excel files, the data is included as one single set). Data fields defined in this component will be placed at the top of each printed page.

NOTE: This component may not be available in single-page invoice templates.

Fields

Similar to the **Header**, this component contains global data for the object to be processed (tariff, invoice, etc.). Data fields are split between the **Header** and **Fields** simply for convenience, with the **Header** containing the main parameters, while **auxiliary** parameters are included in **Fields**. For instance, templates for tariff upload/download contain specialized **Fields:** off-peak period definition, free seconds, connection fee, and so on.
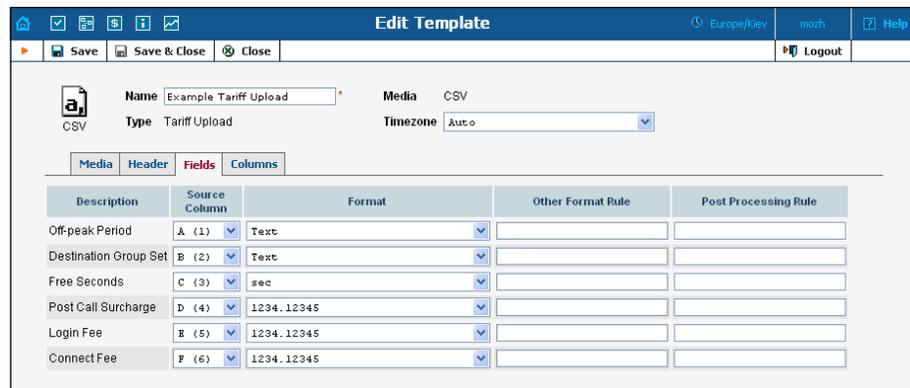
Figure 1-6

Columns (repeating groups)

This is the most important component, since it describes the variable portion of the data. For example, rate entries all have the same structure and format (phone prefix in the first column, country name in the second, price in the third, and so forth). However, there are many such rows in

the file, and each row is an independent object. Thus, you define how one single row of data is to be formatted, and this will then be applied to all rows in the file.

Footer

Contains additional data about the object; for example, invoice totals.

Page Footer

Available only for printable forms; data fields to be placed at the end of each printed page.

Groups

See the *Template groups* section.

Template parameters

Template type

The template type defines the intended use of the template:
- Whether it is for data import (***upload***) or exporting data from PortaBilling (***download***)
- What kind of data is involved (invoice data, tariff data, destinations, etc.)

Examples of template types are "Tariff Download" and "Destination Upload".

Template media

This defines the file format to be processed, for example .CSV (Comma-Separated Values), .XLS (Microsoft Excel), or a printable form (a specific HTML, which might be converted into a PDF file).

Template time zone

Normally, when people specify a point in time, they name the day of the month, the month, the year, the hour and the minute, and, optionally, seconds as well. However, in order to be absolutely precise, this definition should also include a time zone.

Let's assume that your vendor sends you a data file containing his new rates and, for each rate, the date on which it becomes effective. This is fine; but is **01-May-2003 00:00** in the Pacific Standard time zone, or the Central European one? Obviously, time zone misinterpretation can render

the rest of the data invalid. In most cases, specification of a time zone is not included in the date itself, but is defined externally.

For example, your vendor might tell you that all of the given times are in Eastern Standard Time. Or, there might be a note in the file saying this. In any case, the template engine needs to know what time zone is to be used. This is specified in the **Template time zone** parameter. If the parameter is set to "Auto", this means that the time zone of the user performing the upload is to be used. This is convenient when working with your own tariffs, but is not usually acceptable when uploading tariffs from vendors.

NOTE: If the date includes a time zone (for example, **2003-04-10 12:00:00 Australia/Sydney,**) this time zone will take precedence over the template's time zone. However, make sure in this case that the time zone specification is both supported and unambiguous. For example, **EST** could mean Eastern Standard Time in both the US (GMT -5) and Australia (GMT +10) – quite a big difference!

Media parameters

Media parameters define the data layout within the file, where the data for individual components should be located in the file, and whether they are present at all.
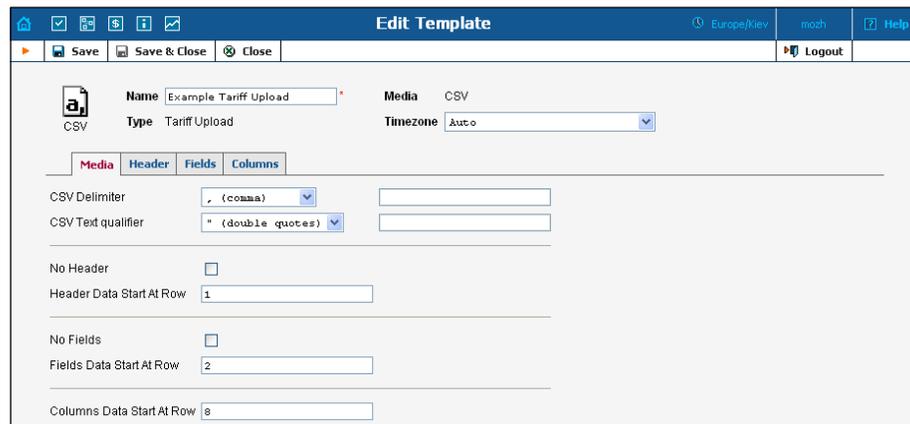
Figure 1-7

Here is a sample data file that matches this template. Note that "…Data start at row" means the first row with actual data, not including the title. For example, line 7 contains titles for column data, and so 8, the number of the first row with rate data (4066667….), appears in **Column Data Start At Row**.

Figure 1-8

Template data fields

Data fields describe a single element of data, such as "invoice number", "price" or "company name". For each data field in the template you specify the following:

Figure 1-9

Source column (for upload templates)

Source columns define which column in the original data contains the desired value. If data for this field is absent from the file, choose **None**.

Multiple data fields can be used for the same source column. For example, if your input file has a single price for calling a specific destination, both **Price 1** and **Price N** will use it as the source column.

Destination column (for download templates)

Destination columns define where to position the result value.

NOTE: This option is not available for download templates with printable form media, since in that case you will specify where you would like the value to appear using Layout Designer.

Format

A format describes data fields so that:

- they may be converted from that format to a unified representation in the database during upload, and
- they may be converted from the database (PortaBilling default) format to a specific format during upload.

For example, if you choose format "011**E.164**" for the "Destination" data field, then the template engine will know that destinations in the input file data are prefixed by "011". Therefore, "011" would be stripped from the destination number "**01116041234567**" during upload, and entered into PortaBilling as "**16041234567**". Exactly the opposite will happen during download: "**011**" will be prefixed to the number "**16041234567**", so the number in the file will be "**01116041234567**".

In order to provide better quality control of data, PortaBilling supports several data types. These define what kind of data appears in the data field, as well as possible ways to process it. Currently supported data types include:

- Destination (phone number).
- Date/time
- Number
- Time interval (number of minutes/seconds)
- Text
- Boolean (yes/no)

A data type is predefined for each data field, so you do not need to specify one. For each data type in PortaBilling there is a set of default formats, so that you will usually not have to invent anything from scratch. If you encounter a data format for which there is no suitable data type in PortaBilling, you may choose "Other". You will then be able to specify a custom format rule.

"Other" format rule

If none of the PortaBilling default formats suit your needs, you can choose "Other" and enter your own formatting rule. See below for a description of how a format rule works for data fields of a particular type.

Number

For numeric data fields (for example, "Price 1" or "Connect fee"), the value you enter in the **Other Format** field is considered to be the format specification **###,###.##**, where:

- A **#** sign or the digits 0-9 represent one digit of the input or output number.
- The first character from the *right* which is not a digit or # (e.g. . in #,###.00) specifies a decimal separator.
- The first character from the *left* which is not a digit or # (e.g. , in #,###.00) specifies a group separator.
- The number of characters after the decimal separator specifies the precision of the number. However, # indicates no zero padding, and 0 indicates zero padding.
- If the number has too many digits for the format specification, it will be rounded to the number of decimal digits specified in the format.
- A group separator (if present) specifies which character should be used to split groups of 3 digits in the number. If none is specified, then no group separation will be done.
- It is sufficient to specify a group separator only once, even if your real numbers will be large and contain more than one digit group. All digits to the left of the decimal point are always shown, regardless of how many of them are actually shown in the format.

This is important mainly for download templates, where the format or precision you use to present numbers in documents differs from the format in which data is stored in the database. For upload templates, number format specification is mainly used to handle non-standard group or decimal separators. See the table below for examples of formats for converting numbers.

Format	Number	Result	Comment
#,###.##	1.6	1.6	No zero padding
#,###.00	1.6	1.60	Zero padding
#,###.00	34567.6	34,567.60	Digit groups
####.00	34567.6	34567.60	No digit groups
#.#	2.382	2.4	Rounding
#.#	2569.31	2569.3	Significant part is never truncated
#.###,##	2569.31	2.569,31	German format

# ###,##	7892569.31	7 892 569,31	Yet another custom format

Table 1-1 - Examples of number formats.

Date and Time

This data type is used for fields containing either both date and time (e.g. **Effective from**) or just the date (e.g. **Invoice date**). Please note that when importing dates from an unusual format you can use the Time::ParseDate module to take care of all the dirty work. This module is able to examine a data string and find bits of data regardless of how they are placed or separated there. In order to use the Time::ParseDate module for parsing dates, choose "Auto" as the format.

Unfortunately, in some situations even Time::ParseDate is unable to properly identify the date. For example, the string "02-01-2003" could either mean February 1st or January 2nd. In such situations, an exact format specification is required. The "Other" format uses a syntax very similar to the one used in applications like Microsoft Excel. Its elements are the following:

- **YYYY** – Year (four digits, e.g. 2005).
- **YY** – Year (short format, e.g. 05). Note that this format can lead to potential data interpretation problems, since "10" could be either 2010 or 1910. The PortaBilling template engine considers the year values 00-30 to be in the 21st century, and 31-99 to be in the 20th century.
- **MM** – Number of the month (01…12).
- **MON** – Abbreviated name of the month (e.g. Jan). Only English names of the months are currently supported.
- **MONTH** – Full name of the month (e.g. January). Only English names of the months are currently supported.
- **DD** – Day of the month with a leading zero for 1 through 9; for example, 01, 06, 14, 25.
- **D** – Day of the month without a leading zero for 1 through 9; for example, 1, 6, 14, 25.
- **HH** or **HH24** – Hour (24-hour format).
- **HH12** – Hour (12-hour format).
- **AM/PM** – AM or PM indicator.
- **MI** – Minutes.

NOTE: Very often people make the mistake of using **MM** as a format for minutes. This is incorrect, as **MM** is a format for month. Thus, **HH:MM:SS** will clearly not work correctly.

- **SS** – Seconds.
- **TZ** – Time zone name (e.g. CET).

- **XXX** – Any combination of non-space characters; used in formats for parsing dates to skip a variable part of the date, for example, the name of a weekday (see examples below in Table 1-2).
- All other symbols in the format (spaces, delimiters like **/**, **-**, **:** or **;** and other characters) represent themselves. So they are placed as-is in the output date string, and must be found in the input string exactly as they were entered.

Let's take a look at some examples for importing dates into PortaBilling:

Date/time string	Description	Correct format specification
07/04/2003	July 4th 2003, US date format	MM/DD/YYYY
04-07-2003	July 4th 2003, European date format	DD-MM-YYYY
07-04-03	July 4th 2003, US date format, short year format	MM-DD-YY
11:05 am	Time, 12-hour format	HH12:MI AM/PM
11:05	Time, 24-hour format	HH:MI
18:58:00	Time with seconds	HH:MI:SS
2003-07-04 14:00:00	Date and time, ISO format	YYYY-MM-DD HH:MI:SS
07/04/03 2:00:00 pm	Date and time, US format	MM/DD/YY HH12:MI:SS AM/PM
1-May-2002 18:52	Date and time, abbreviated month name	D-MON-YYYY HH:MI
Sun, 16 Mar 03	Date and time, skip weekday name	XXX D MON YY
12th of March, 2003	Date with "as is" elements	Dth of MONTH, YYYY

Table 1-2 - Examples of date and time formats for upload.

The following table gives examples for using templates to convert values in the PortaBilling database to the desired format.

Date/time in PortaBilling	Format specification	Resulting value
01-May-2003 12:34:56	MM/DD/YYYY	05/01/2003
01-May-2003 12:34:56	DD-MM-YYYY	01-05-2003
01-May-2003 12:34:56	MM-DD-YY	05-01-03
01-May-2003 12:34:56	HH12:MI AM/PM	12:34 pm

01-May-2003 12:34:56	HH:MI	12:34
01-May-2003 12:34:56	HH:MI:SS	12:34:56
01-May-2003 12:34:56	YYYY-MM-DD HH:MI:SS	2003-05-01 12:34:56
01-May-2003 12:34:56	MM/DD/YY HH12:MI:SS AM/PM	05/01/03 12:34:56 pm
01-May-2003 12:34:56	D-MON-YY	1-May-03
01-May-2003 12:34:56	D of MONTH, YYYY	1 of May, 2003
01-May-2003 12:34:56	YYYYMMDD HHMISS	20030501 123456

Table 1-3 - Examples of date and time formats for download.

Other data types

For other data types (e.g. **Destination, Text** or **Interval**) there are no specific format elements. The "Other" format rule is regarded as a Perl macro; thus it is evaluated, and the result of the evaluation is considered a new value. For a detailed description of Perl macros, see the Post-processing rules section below. The original string value is received as a parameter, and the converted value should be returned in the required format.

The following table provides a description of which type of output value is required for other format rules, depending on the data field type.

Data field type	Other format rule in	Output value
Destination	Perl code	string
Number	Number format	N/A
Boolean	Perl code	string ('Y' or 'N')
Interval	Perl code	number
DateTime	Date/time format	N/A
Text	Perl code	string

Table 1-4 - Output value type for other format rules.

Post-processing rules

Sometimes you will need to perform a very specific conversion which is not possible using the standard PortaBilling formats. In this case, you can use post-processing rules. A post-processing rule is a piece of Perl code which is executed at run-time, and the value it returns becomes the final value for a field. This is similar to formulas or macros in Excel, giving you a virtually unlimited ability to change your data as you need to. There are a few requirements for writing post-processing rules:

- A processing rule is Perl code, i.e. one or more Perl operations separated by semicolons.

- The input value (to be processed) is in the **$value** variable.
- The final result is the value returned by the last expression in the list.

TIP: One of the most useful things post-processing rules can do is to create new data which is not based on the input data. This is required, for example, when certain data fields (e.g. **Interval 1** and **Interval N**) are absent from the input file. If you know that **Interval 1** is 30 seconds, and **Interval N** is 6 seconds, you can simply assign these values as constants in the post-processing rule.

Let's take a few examples:

Post-processing rule	Input data	Result
Remove all leading white space from the string		
`$value =~ s/^\s+//;`	___Aruba___	Aruba
Remove all leading and trailing white space from the string		
`$value =~ s/^\s+//;` `$value =~ s/\s+$//;`	___Aruba___	Aruba
Add 1 before the phone number, but only if it does not start with 0		
`$value =~ m/^0/ ?` `$value : '1'.$value;`	6041234567	16041234567
Take only the part of the input string following the hyphen		
`my @a = split('-',` `$value, 2); $a[-1];`	Norway-Oslo	Oslo
Assign a constant value to the field		
`$value = 30`	None	30
30	None	30
Replace all hyphens with spaces and remove all quotes		
`$value =~ s/-/ /g;` `$value =~ s/\"`	Germany-'Berlin'	Germany Berlin
Use an external (custom) module to translate English country names into Czech		
`use Czech::Translate;` `transl_data($value);`	Austria	Rakousko

Table 1-5 - Examples of using post-processing rules.

NOTE: This manual does not cover details of programming in Perl. There are excellent books on this subject, for example:
- **Programming Perl** by Larry Wall, Tom Christiansen, Jon Orwant (ISBN 0-596-00027-8)
- **Learning Perl** by Randal L. Schwartz, Tom Phoenix (ISBN 0-596-00132-0)
- **Perl Cookbook** by Tom Christiansen, Nathan Torkington (ISBN 1-56592-243-3)

Values returned by a post-processing rule should have the same type as the input value it receives. The following table provides information about types of input values for post-processing rules, depending on the data field type.

Data field type	Input value ($_)
Destination	String
Number	number
Boolean	string ('Y' or 'N')
Interval	number
DateTime	unixtime
Text	string

Table 1-6 - Input value type for post-processing rules.

Template groups

Concepts

Template groups are necessary for performing sorting/subtotals on large sets of data. For example, you might need to make a list of all phone calls, with a subtotal for each country.

Page 2 of 2							
		Call Detals Records By Country					
CANADA							
3804424471821	160452152771	CANADA	Quebec	2004-02-12 08:08:25		38	0.10
3804424471825	160452152771	CANADA	Alberta	2004-02-12 08:09:07		3	0.49
3804424471825	160421971327	CANADA	British	2004-02-12 08:09:13		31	0.26
3804424471825	160421971325	CANADA	Ontario	2004-02-12 08:09:54		3	0.04
TOTAL By CANADA						2	0.88
CZECH REPUBLIC							
3804424471821	420608076502	CZECH	Mobile	2004-02-12 07:23:07		14	0.75
TOTAL By CZECH REPUBLIC						0	0.75
UKRAINE							
3804424471825	380688763541	UKRAINE	Mobile	2004-02-12 06:20:29		66	0.36
3804424471825	380442447182	UKRAINE	Kiev Region	2004-02-12 07:09:35		1	0.95
3804424471825	380503425341	UKRAINE	Mobile	2004-02-12 07:10:16		20	0.24
3804424471825	380664729846	UKRAINE	Mobile	2004-02-12 07:14:25		2	0.13
3804424471825	380672345123	UKRAINE	Mobile	2004-02-12 07:15:07		21	0.15
TOTAL By UKRAINE						2	1.83

Figure 1-10

Group

Each group is a portion of data which has an identical value for some parameter or parameters. All of the input data will be sorted according to

groups, and rows within the same group will be presented together. In the example above, three groups are visible, each of them containing all calls made to a certain country. Groups can also be nested; for example, the group "By Country" will contain several sub-groups "By Prefix". These are sorted in the list of groups according to their include order, so that the outermost group is on top, and the innermost at the bottom.

Group Break

This is a parameter which defines which group the data should belong to. All of the available data are sorted according to this field. In the example above, the list of all calls is sorted by the name of the destination country (so that calls made to **Slovenia** appear before calls made to **Ukraine**). Next, rows which have the same group break expression value are joined in the same group, as seen in the example above for all calls made to Slovenia, Ukraine or the US.

Normally, you will just choose one of the available columns for breaking the group (in the example above, this is the "Country" column). You can also create a custom group break, specifying any valid SQL expression.

Group Header

This is a template component which is inserted into the document immediately before the group data. Besides static text, variable fields (e.g. "Destination") can also be used. In the example above, this is the row containing the name of the country.

Group Footer

Similar to the group header, this element is placed immediately after the last row of the group. It can consist of static text and variable fields. In the example above, this is the row containing "TOTAL by" and the name of the country, as well as the total number of seconds and amount for the country.

Group Fields

These variable elements are very similar to data fields in the **Columns** section of a template (and are actually based on them). You can alter their parameters (such as format or post-processing rules) in exactly the same way as you would for standard elements of the **Columns** section.

Group totals

Since one group may consist of many rows of data, what would be the value in the group fields for the header/footer? The fields in the group header have the value of the first row in the group. You can specify how you would like to populate the value of fields in the group footer. Possible pre-defined options include:

- Summary (sum of values for the corresponding column across all rows)
- Count (number of rows in the group)
- Last value (value of this column in the last row)

You can always write your own custom total rules. These are very similar to post-processing rules, in that you can specify any Perl expression. The current value of the column is given in the **$value** variable, the current value of the total is given in the **$total** variable, and the result returned by the expression becomes the new total.

In the example below you see what a list of calls might look like if you were to use two groups:
1. Break group by country (so that each group contains all calls made to the same country).
2. Break group by destination. This group exists within the first group, so it contains all calls made to a specific destination within a certain country.

Figure 1-11

Things to remember

Destination upload

Why can't I use custom templates for destination upload/download?

There is only one format available for uploading destinations at the moment, since:
- The format is fairly simple.

- You almost never need to upload new destinations alone; normally, you will need to input new rates in the system which requires the new destinations, and in this case the PortaBilling template processor will provide you automatically with a list of new destinations in the proper format.

However, a functionality for creating custom templates for destination upload or download is available, and may be included in future releases.

How to fill in country information for a destination

For each destination entered into PortaBilling, it is necessary to provide information on which country (and, optionally, administrative subdivision in that country) the destination belongs to. This will help you keep better track of rates for a specific country, and will also be used to ensure a correct description in call details and statistics for you and your customers. PortaBilling comes with a complete list of countries registered with the International Standards Organization, providing each country's ISO code (e.g. **US** for United States of America, **CA** for the Canada, and so on). When a new destination is manually entered using the PortaBilling web interface, you can choose the country from a drop-down select menu. When importing many new destinations at once, country information must be provided for each in the form of the two-letter ISO code. Why not the country name? Unfortunately, country names can be very ambiguous. For example, *UK*, *United Kingdom*, *Great Britain*, *Britain* and *England* are all synonyms for the same country (which could also could be called *The United Kingdom of Great Britain and Northern Ireland*), and you never know which version of the name your vendor will decide to use. Moreover, country names in local languages (e.g. *Velka Britanie* in Czech), spelling errors (e.g. *Untied Kigndom*), punctuation marks, and the like further complicate the matter. This is why PortaBilling uses country codes instead, since they define exactly which country is meant.

Does this mean that you now have to fill in country codes for several hundred missing destinations manually? No, there is a better way!

TIP: When uploading destinations with empty country codes, PortaBilling will try to find country information based on data which is already in the database. For example, if the prefix **61** already exists in the database, and is assigned to Australia, then when you upload a new destination with prefix **61881** PortaBilling will automatically fill in the country information for you.

NOTE: Of course, in order for this method to work effectively you need to have an initial set of destinations in the database already. PortaBilling comes with a default set of almost 3,000 destinations, which covers most of the countries of the world and popular destinations in those countries. You can download it using the **Get default set** button in the Destinations screen, customize it according to your needs, and then upload it.

Tariff download and upload

There are a few helpful features regarding the upload of tariffs:

New destinations files are created automatically

When you try to upload a rate for a phone prefix which has not been entered in the destinations, the rate will not be inserted. However, PortaBilling will create a list of such missing prefixes in a format suitable for uploading.

Integrity checks during tariff upload

It is a very common error to upload the wrong file (e.g. from a different vendor or in a different currency). In order to prevent this situation, the following improvement has been made in Portabilling100:

- It is possible to include the tariff name and tariff currency in the data. The default templates for uploading and downloading tariffs provided with Portabilling100 include this as well.
- When you try to upload a tariff, the template processor compares the **Tariff name** and **Currency** in your file to the tariff you are trying to upload. If they do not match, the upload will be rejected. This prevents you from making such errors.

This feature only works if you have a **Tariff name** and **Currency** in your file.

You can only upload rates effective immediately or sometime in the future

Since billing is done in real time, a call is billed as soon as possible and a CDR written to the database. So in order to modify CDRs which are already in the database, a re-feed should be used. Therefore, if you are copying data between tariffs and there is an **Effective from** column included in the data, make sure that you replace values in the past with an empty value. An empty value in the **Effective from** column means "immediately".

2. Invoicing

How does invoicing in PortaBilling work?

PortaBilling provides a ready-to-use invoicing functionality. You can design how your invoices will look. Invoices are created automatically based on calls the customer has made; they are stored as a PDF file and can be viewed online or printed. They are also automatically emailed to the customer if he has an email address set up in his properties.

Who is invoiced?

Only customers (both resellers and retail customers) are invoiced, since only they have all the information required for invoicing, such as address, taxpayer ID, and so forth. Individual accounts are not invoiced.

How often?

Each customer has his own billing period, which could be:
- Daily
- Weekly
- Bi-weekly
- Monthly (anniversary or on the first day of every month)
- 30 calendar days

When the current billing period is over (for example, on the first day of the month) a summary of all the calls made by the customer is calculated. Customer call statistics are generated and stored on the server (optionally, they can also be mailed to the customer automatically) and an invoice is created.

Note that the billing period is based on the customer's time zone. For example, if the customer has a weekly billing period and his time zone is Australia/Melbourne, his statistics would cover the period from 24-Mar-2003 00:00 Melbourne time (23-Mar-2003 14:00 GMT) until 31-Mar-2003 00:00 Melbourne time (30-Mar-2003 14:00 GMT). For another customer, who also has a weekly billing period but is in the New York time zone, this billing period will cover calls made from 24-Mar-2003 00:00 New York time (24-Mar-2003 05:00 GMT) to 31-Mar-2003 00:00 New York time (30-Mar-2003 14:00 GMT).

When are statistics available?

Statistics are generated with a sufficient delay to make sure that the billing period for a particular customer (according to that customer's time zone) has ended. Also, statistics calculation takes place only at a certain time of

day (usually during night hours, when there is the least amount of activity in the system). Weekly invoices cover Monday through Sunday. Assuming that your system does statistical calculation from 2am until 8am Central European Time every day, then statistics for those customers whose time zone is in Asia or Europe should be available each Monday morning. For customers with a time zone in the US, the billing period will not have ended yet when statistics are calculated on Monday morning; thus they are created on Monday night and made available Tuesday morning.

There is currently no invoice-on-demand functionality in PortaBilling, i.e. invoices should match customer billing periods, and are generated automatically. However, if you want to include a certain amount (for some special service) in the invoice for your customer manually, this is easy to do: simply perform a balance adjustment. This will modify the customer's balance and also create a CDR record, which will be included with the customer's next invoice.

Invoice total

There are two methods available for calculating the invoice total:

All charges during the period

This is the simpler method (and the only one which was available prior to PortaBilling100 Maintenance Release 9). The invoice contains all of the call charges for the specified period, with the invoice total equaling the sum of these charges. Balance adjustments, refunds and payments made by a customer via the online payment system are not included, since these refer to transactions which have already been made and will not occur again. So, for example, if during August a customer makes calls for $120 and pays $100 for his services in July on August 10th, his invoice will show $120. Of course, the customer's payment will be recorded in the system, and will affect his balance.

Including the previous balance

The invoice total will be calculated as:
- Previous invoice total
- + Sum of all charges during the period (call-related or manual charges)
- - Sum of all payments or refunds made during the period

So, taking our previous example: if the customer's invoice for July was $110, and he paid $100 and made calls for $120 during August, his total on the August invoice will be $110 + $120 - $100 = $130

Invoicing modes are controlled by the `New_Style_Invoice_Calculation` switch in the `porta-admin.conf` file.

What the invoice is based on

For a reseller, the call charges on the invoice are based on the sum total of all his CDRs in a given period.

For a retail customer, the call charges on the invoice are the sum total of all CDRs (call charges and maintenance fees) for this customer's credit accounts (debit accounts are prepaid, and so since they have been already paid in full there is no need to invoice these calls).

Invoice template

The invoice template is a specially formatted HTML document, which is created using the PortaBilling Layout Designer (see the following chapter for a detailed description of the Layout Designer). It defines how the invoice should look, with information on fonts, colors, pictures, and placement of the components. When you view the invoice on the web interface, or when the invoice is automatically emailed to the customer, the template engine takes the invoice template and simply fills in the data fields (such as **Header.Invoice Number**) with their real values (for example, **1001** as the invoice number). The result of this processing could be an HTML document or a PDF file.

Requirements for invoices vary greatly from country to country (and even within the same country). The default invoice template supplied with the system is suitable for British Columbia, Canada, and is provided as an example only. Be sure to create your own invoice template and assign it to your customers.

Two predefined invoice templates are supplied with PortaBilling: "Single-page invoice" and "Multi-page invoice". What is the difference between them?

Single-page invoice

This is a traditional-style invoice, similar to the invoice template in the older version of PortaBilling. It consists of a single page with an invoice header (your company name, customer name, etc.), invoice fields (invoice number, invoice date), and invoice footer (subtotal, total).

Multi-page invoice

This template's first page is identical to the single-page invoice template. In addition to this, it may contain multiple pages with details of calls related to the invoice. You can use the button in Layout Designer to switch between the main page and the call details page. All of the calls made by a customer can be included, or only summaries.

Billing, invoicing and taxes

Does PortaBilling perform billing with or without taxes? Actually, PortaBilling is a rating engine, so it all depends on how you use it. PortaBilling does all the calculations based on rates you enter into the system, as well as tariff parameters and call duration. So, in this sense, it makes no difference whether the price includes tax or not.

Thus, there are two scenarios available for dealing with taxes, and each has its own advantages and drawbacks.

Calculating CDRs without calculating tax

If you enter rates into PortaBilling without tax, CDRs are calculated without tax. When you create an invoice for your customer, you can include all the necessary tax calculations there.

The advantage of this method is unlimited flexibility: you can implement support for any taxation scheme you need. Its big disadvantage, however, is that, since all of the CDRs exclude taxes, customer balances will also not include tax. This makes it difficult to match customer balances in PortaBilling with actual balances, since the amounts customers pay you include tax. Also, this method does not work well with prepaid cards: when a customer buys a $10 calling card, he expects to spend $10 on calls and is interested in what his price per minute will be, without complications like taxes.

Calculating CDRs with taxes

If you enter rates into PortaBilling including taxes, you will get a "final" price, and the CDRs are calculated accordingly (with tax included). In this case, the balances are correct, and prepaid products work fine. The only complication is providing proper tax information on the invoice (if this is required), since now it has to be done backwards (i.e. you have to deduct tax from the amounts).

This is the recommended method, as it provides accurate billing for both prepaid and postpaid products. The default invoice template supplied with PortaBilling provides an example of how to set up invoicing in this way.

Back-calculating taxes based on the invoice total

If you wish to include tax in your rates, you need to show proper tax information on the invoice. Five data fields are available for different types of taxes (for example, GST and PST in Canada), which you can fill

in using different formulas. These all receive a total invoice value as their input value; then you can calculate the actual tax value using your own formula in a **post-processing rule**.
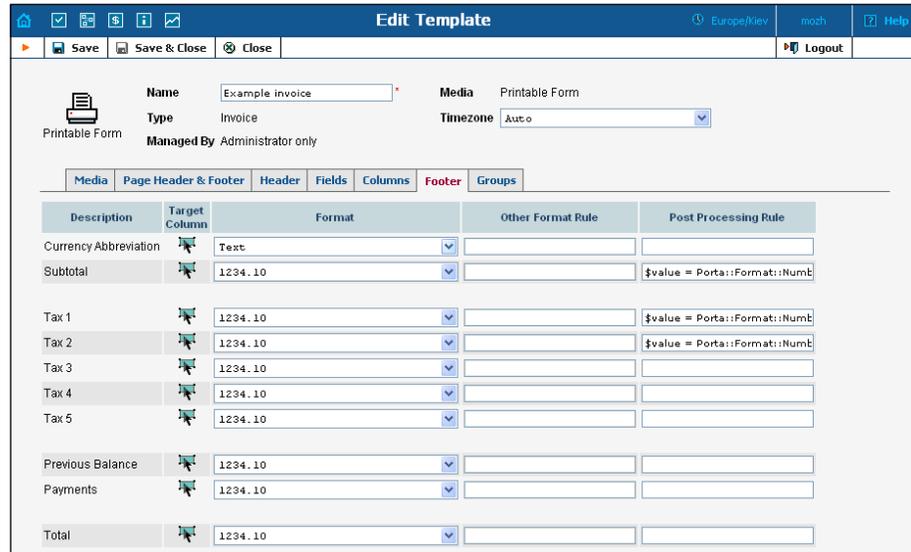
Figure 2-1

The PortaBilling template library has a function which can be used to easily calculate the amount of certain types of tax. The `Porta::Format::Number::backcalcTax` function consists of three parameters:

- Total amount
- Total tax percentage
- Percentage value of this tax type

Let's take the example of British Columbia, Canada. You would need to include two types of tax: GST (7%) and PST (7.5%). Then, for the **Tax1** data field, you just have to assign this post-processing rule:

```
Porta::Format::Number::backcalcTax($value, 7 + 7.5, 7.0);
```

while for **Tax2** the post-processing rule would be:

```
Porta::Format::Number::backcalcTax($value, 7 + 7.5, 7.5);
```

As you can see, this is quite simple. Instead of writing `7 + 7.5` you could have written `14.5` – but why do the calculations if the template processor can do them for you?

Finally, it is only necessary to put **Footer.Tax1** and **Footer.Tax2** fields on the invoice where you want them to be, and label them appropriately.

Figure 2-2

3. Layout Designer

Layout Designer is a visual (WYSIWYG) tool for creating document templates. It is implemented using only Javascript and DOM HTML extensions, so it will work in the MS IE or Mozilla web browser, and no installation of extra components on the user's computer is required.

When you design a template in Layout Designer, you define how the document should look: the location of components, pictures, fonts, and colors. You also place the data fields on the document. These are containers for real data, defining where the data element should appear.

For example, on the invoice template you put an **Invoice.number** data field in the place where you would like the invoice number to be printed (top-right corner, in the example below). When the actual invoice is generated based on this template, **Invoice.number** will be replaced by the actual invoice number (e.g. **557**). Make sure you insert the data field using the pop-up menu, because if you simply type **Invoice.number** it will be regarded as a text string, and so will appear on the invoice exactly as you have typed it.

NOTE: During design, data fields are represented on the screen by sample data, e.g. **25** for the invoice number. This data is provided only for your convenience, and is used only in design mode. All sample data are shown in <u>over lined font</u>, which makes it easy to recognize.
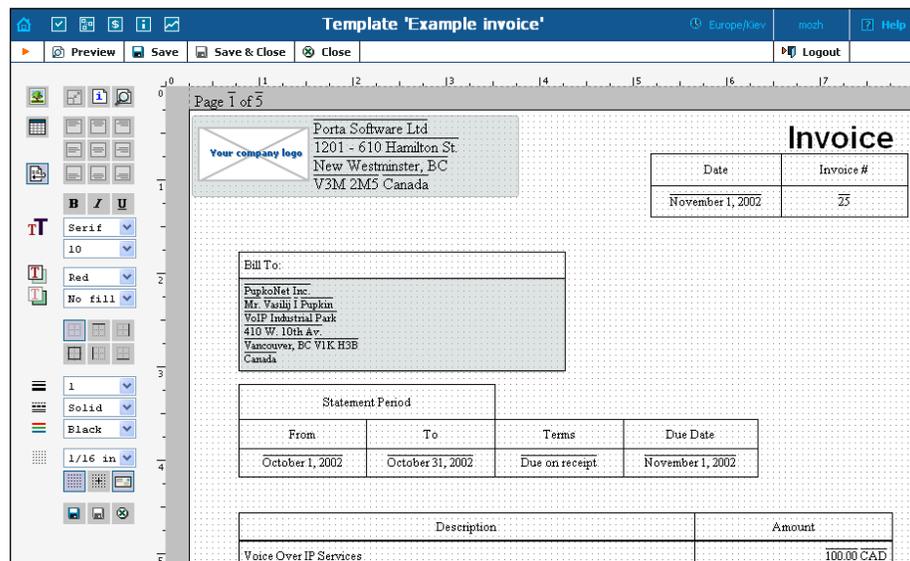
Figure 3-1

Layout Designer consists of two main parts:
- Toolbar (on the left). It contains buttons and controls for manipulating content in the editor window.
- Worksheet (on the right). It contains the document itself.

General document properties

Page setup

In order to access page setup properties, right click on the empty background of a template window (i.e. not on an object such as a table or image).
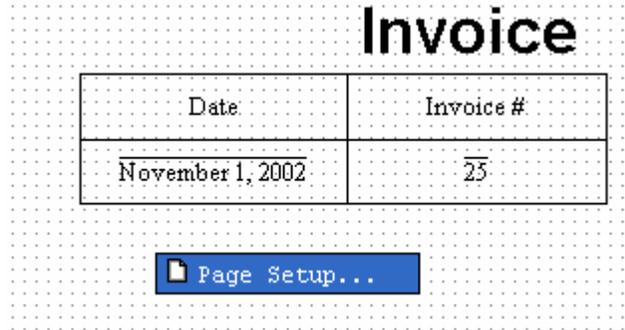
Figure 3-2

Choose the "Page Setup" item.

Figure 3-3

You can choose the paper format, margins, and the position of envelope windows. Consult the online help for a detailed description of different envelope styles.

Grid properties

The grid helps you easily adjust objects in the template. There are a few buttons on the toolbar which control the grid's behavior:

- - Turn the grid on or off (by default it is on).
- - Turn snap to grid on or off. When snap to grid is on, object edges will always be on one of the grid lines, so you cannot place an object between the grid lines when dragging or resizing it.

- The select menu allows you to choose the size of grid cells.
- [image] - Shows or hides the envelope window on the template.

Working with images

You can add your own images to the template. These images will then be stored on the server for use by anyone browsing documents created from the template. To add a new image:

- Click on the empty background of the template (i.e. no other object, such as a picture or table, should be selected). The **Add picture** button will become enabled.

- Click on the [image] **Add Picture** button.
- The **Open File** dialog will appear. Choose a picture on your local disk. Any picture format which is supported by popular web browsers can be used (.JPG, .GIF, .PNG).
- Click **Open** and the picture will be transferred to your template. You can now move or resize it.

Placing the data elements

The main object of the Layout Designer is the table, a placeholder for all inserted information. A table contains text or data fields which otherwise could not be put into the template. To add a new table to a template:

- Click on the empty background of the template (i.e. no other object, such as a picture or table, should be selected). The **Add Table** button should become enabled.
- Click on the **Add Table** button.

A new table will then be added to the template. This is a very basic table, consisting only of a single cell. However, you can change it. To move the table to the desired place, simply drag it there (click on the table and move the mouse without releasing the mouse button). To perform advanced operations on the table (e.g. adding a row), right click on the table so that a menu appears.
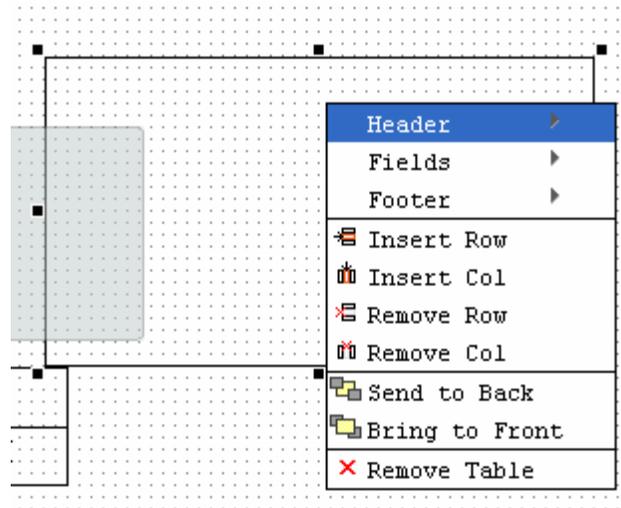
Figure 3-4

A table or a cell within a table can be moved and scaled by dragging the cell handles. Double click a table or a cell to enter cell editing mode, where you can type in or edit the cell's text. To change the attributes of the cell's text, make sure the cell is selected (8 black square handles are visible) and set the text attributes in the toolbar on the left.

Changing a cell's text attributes in the toolbar will affect all text in the current cell. To change the attributes of part of the text, enter cell editing mode (double-click), select part of the text, and use the keyboard shortcuts listed below:

- Ctrl + B - Bold
- Ctrl + I - Italic
- Ctrl + U - Underlined

To insert data fields into the cell, do the following:

- Right-click on the cell where the data field should be inserted.
- In the top part of the menu, click on the component name to which the data field belongs (e.g. **Header**). A submenu with a list of all data fields in this component will appear.
- Click on the name of the data field you would like to insert.

You can always go into cell editing mode and add text before or after the data field. Also, by removing all of the over lined data field name, you can completely delete the data field from the cell.

Changing the text attributes

You can apply multiple formatting styles to text elements in a table (this includes both static text and data fields), namely:

- Text alignment within the cell,

- Text font and font attributes, text color,
- Table cell borders and border style.

Consult the online help for a complete list of all possible text operations.

4. Practice Session

Hands-on session

In the following chapter we will try to deal with situations very similar to those you might encounter. Of course, the details in your case will likely be different; here we will be covering some general techniques.
In order to achieve better results in learning to use PortaBilling templates, we encourage you to actually perform the operations described below on your system as you read this chapter. You may download sample tariff data from **http://www.portaone.com/resources/ExampleTariff.csv**.

Importing rates from your vendors into PortaBilling

Input data

Your vendor PowerTel sends you a list of new rates for termination. You need to import them into PortaBilling.

Figure 4-1

Input data analysis

First of all, let's try to identify if the file is suitable for upload at all, what structure it has, and where the data is located.

- The file is in Microsoft Excel format.
- There is no tariff name, currency or description in the file. In other words, there is no **Header** data.
- There is no information about off-peak periods, connect fees or other **Field** parameters, either.
- There is data for the rates. This is located on the worksheet called **Sheet1**. Each rate is in a separate row, and the two most important parameters, phone prefix and price, are present. The first row with data is row **2**. The missing information (for example, duration of billing intervals) should be obtained by some other means (for example, contacting the vendor directly).

Therefore, the file contains the necessary data, and may be imported.

Create a template

From the main menu, choose **Templates** to open the template management screen.
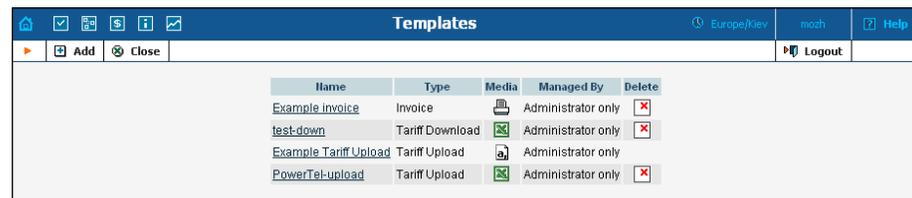
Figure 4-2

Click **Add new** and fill in information about the new template. The template type is **Tariff upload**, the media is **XLS**, and the template is created as a copy of the **Tariff upload XLS** (default PortaBilling template). Press **Save**.
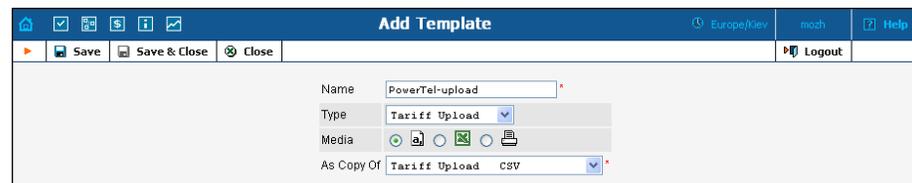
Figure 4-3

Template parameters

After the new template has been created, you need to fill in media parameters and formats for the individual data fields.

Media parameters

Here the media parameters would be:

- **No Header**, since there is no header data in the file. This completely disables the component; so the values in **Header Worksheet** and **Header Data Start At Row** are ignored.
- **No Fields**, since there is no field data in the file.
- **Column Worksheet** is "Sheet1" and **Column Data Start At Row** is "2".
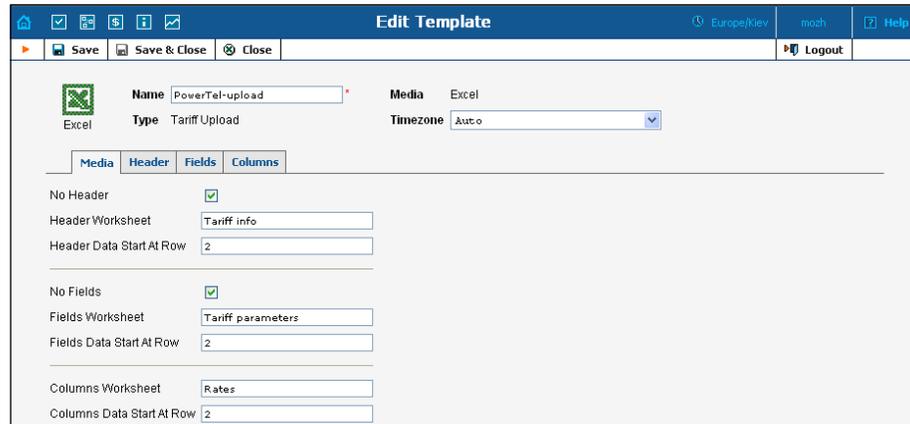
Figure 4-4

Header

Since there is no header data in the file, and you activated the **No Header** switch in the media parameters, this tab can be ignored.

Fields

Since there is no field data in the file, and you activated the **No Fields** switch in the media parameters, this tab can be ignored.

Columns

Figure 4-5

- **Destination description** – Data is in the first column, so choose **A (1)** as the source column. No extra formatting is required. **Hint**: In order to extract the location description ("Tirana") from the destination description "Albania-Tirana", the following post-processing rule can be used: `$value =~ s/^.+\-//; $value;`

- **Destination** – Data is in the second column, so choose **B (2)** as the source column. Since phone prefixes are already in the E.164 format, choose **E.164** as the format; no post-processing rule is required.

- **Interval 1** and **Interval N** – There is no data for these in the file, so you should choose **None** as the source column. Let's assume that the vendor has informed you that he uses 1-second billing intervals during peak hours and 30/6 billing intervals otherwise. Then you need only enter these values as constants in the post-processing rule.

- **Price 1** and **Price N** – Since, in this case, the price (per minute) for the first interval and the rest of the intervals are the same, you use the same source column **C (3)**. Use the format **12345.12345**, since the numbers have "**.**" as a decimal point.

- **Offpeak Interval 1** and **Offpeak Interval N** – Fill these in with constants, similar to what was done for **Interval 1** and **Interval N**.

- **Offpeak Price 1** and **Offpeak Price N** – Since the price is the same for on-peak and off-peak, use the same source column and format as for **Price 1** and **Price N**.

- **Forbidden**, **Discontinued**, **Hidden** – There is no data for these data fields in the file, so choose **None** as the source column.

43

- **Effective from** – Choose **None**, so that the rates will be active immediately.
- **Formula** – Choose **None** (since we assumed that the vendor uses 30/6 billing increments, we do not need a custom rating formula).
- **Preference** - a custom preference for routes of this carrier. If you set a value of the Source Column parameter to None, a default preference of 5 will be assigned.

After you are done with the template parameters, click **Save&Close** and then go to the tariff management screen.

Create a tariff

In the tariff management screen, click **Add New** in order to create a new tariff.
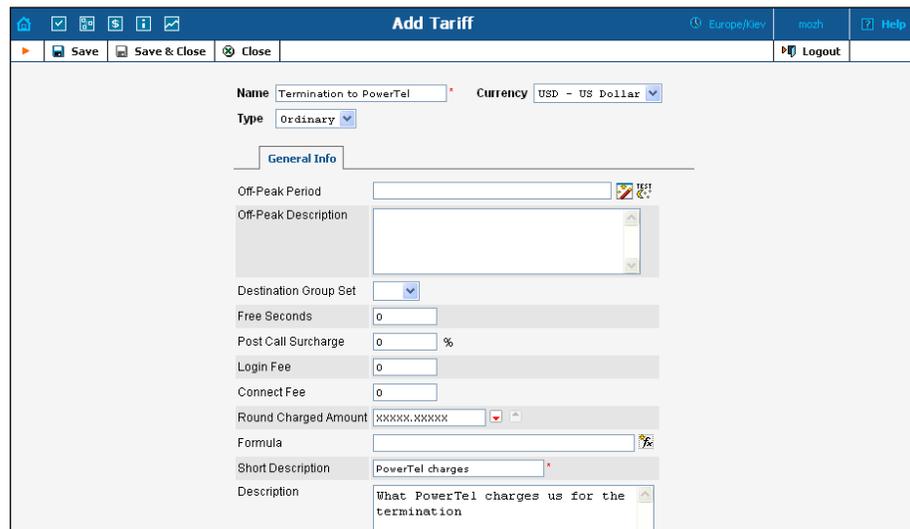
Figure 4-6

After the tariff name and other parameters have been filled in, click **Save**. Now the tariff will be saved and extra tabs will become available. In **Default Upload Template**, choose the template you just created – **PowerTel-upload** – and click **Save**.

NOTE: Always save changes in tariff information by clicking Save before proceeding with the upload of the tariff data.

Figure 4-7

Now you are ready to upload the tariff data.

Upload tariff data

Click on **Upload**, and the upload screen will open. You can type the name of the input file manually, but it is more convenient to use the **Open file** dialog (click on **Browse…** to invoke it).
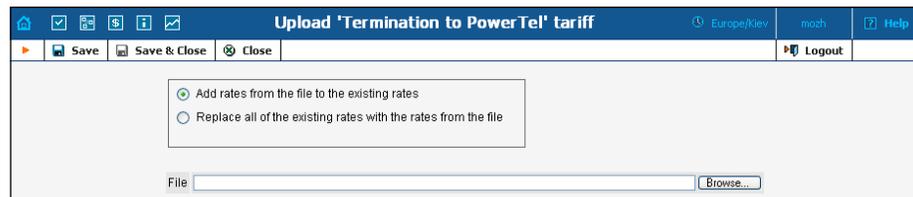
Figure 4-8

After the file name has been entered, click **Save&Close**. This will transfer the file to the server, where processing will start.

NOTE: If you have the input file open in Excel, make sure that you **Save&Close** it first. Excel locks the file so that no other applications can access it.
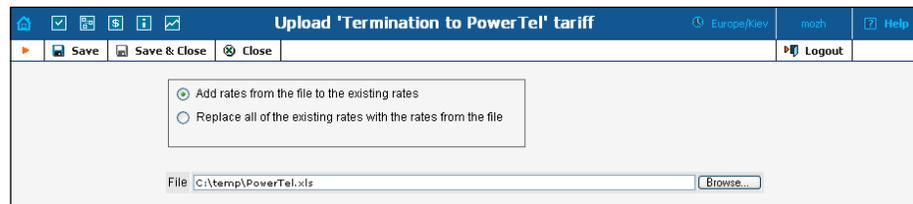
Figure 4-9

Dealing with upload errors

Let's assume that there were some errors during the upload. You will see the following message:

Figure 4-10

Types of errors

There are a few main reasons why a rate will not be inserted:

- There is an error in the input data. For example, the number does not match the format, so that some field is empty. All such rows will be grouped together in a single file to simplify their correction and re-importing. Thus, if you tried to upload 5,000 new rates and 30 of them had an error, you have to edit a file containing just 30 lines and upload only that file again, not the original one.

- A destination is missing. For example, you may have tried to upload a rate for **420602** (mobile phone in the Czech Republic). However, for the purpose of maintaining data quality, each phone prefix must first be defined as a destination. At this step, you can add some extra information associated with the prefix, for example the country name or a custom description. So if destination **420602** does not exist yet, the rate cannot be inserted. In this case, we will have to create a destination first, and then try to upload the rate again.

- Attempting to upload a rate with an "effective from" date in the past. Since PortaBilling performs real-time billing, CDRs are created and accounts/customers/vendors are billed as soon as a call has been completed. So if you try to upload a rate with an effective time one hour before the current time, this will not change anything, since calls made during that hour have already been billed. Moreover, this could only lead to confusion later on, since it might seem that calls were billed using rates which were not effective at the moment. Thus you can only upload rates whose effective time is "immediately" or in the future.

NOTE: You can upload many rates for the same destination within the same import session, provided that their "effective from" dates are different. For example, you can upload a rate as effective immediately, effective 01-Apr-2003, effective 01-May-2003, and so on. It does not make much sense to upload more than one rate for the same destination as effective immediately in one upload session.

Import result in an email message

In any case, you will receive an email containing detailed information about the import results. The email message will look like the one in Figure 4-11. There will be a detailed log explaining why each particular row could not be uploaded. Also, there will be one or more files as an attachment.
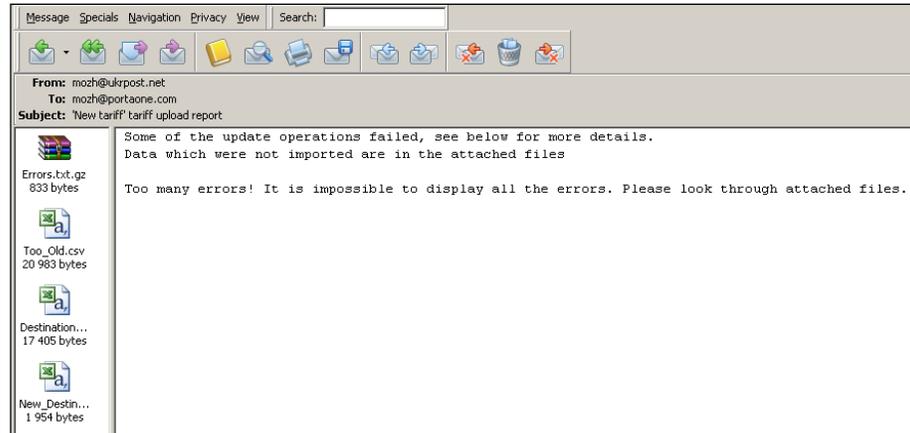
Figure 4-11

- **Data_Error File**: Contains lines from the input file which were not imported due to incorrect data (missing value, incorrect format, etc.).
- **Destination_Unknown File**: Contains lines from the input file which were not imported because the phone prefix for this rate does not exist in the database.
- **New_Destinations File**: Contains a list of phone prefixes which seem to be lacking in the database in a format suitable for destination upload.

The names of these files are defined by PortaBilling, while the extension depends on the actual input file format. For example, if you try to upload a CSV file, then Data_Error and Destination_Unknown will also have a .CSVextension.

Fixing errors in the data

Let's open the Data_Error.xls file and view its contents.

Figure 4-12

Note that rates for the destination in Ukraine were rejected, since there was a formula error in the original file. Also, the price for the Czech Republic has "," (comma) instead of "." (dot) for the decimal point. After

47

you fix these errors, you can upload the file again; it should go smoothly this time.

Fixing the missing destinations

First, let's take a look at the New_Destinations.csv file.

NOTE: Regardless of the input file type (.CSV, .XLS), the file type for destination upload is always a .CSV file, as there is only one template used for destination uploads, i.e. the default one. It is possible to create a user-defined destination upload/download template, which may be applied to data in some future release.

Figure 4-13
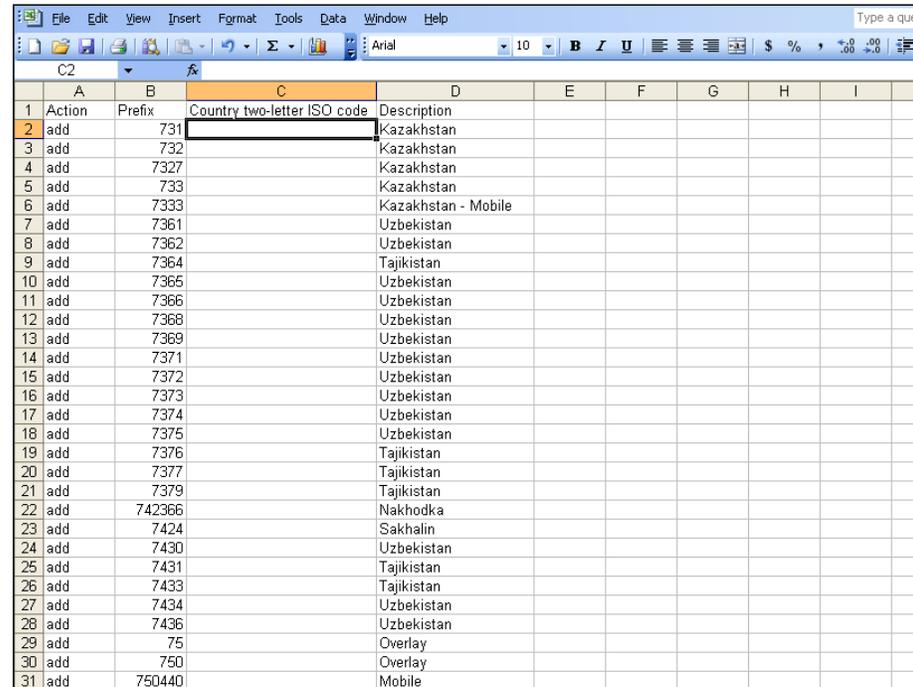
Note that the phone prefix and destination description were transferred from the original input file. "Add" in the **Action** column indicates that you are going to add new destinations. What about the **Country two-letter ISO code** column, which is empty?

Country information for new destinations

As you know from previous chapters, PortaBilling is able to automatically match a country for a phone prefix based on data which is already in the database. In this case, for example, if the database already contains a destination **61** associated with Australia, and you try to upload destination **61881** (line 11 in Figure 4-13), PortaBilling will fill in the country information automatically.

So now you should go to the **Destinations** screen, choose **Upload**, enter the name of the file, and click **Save&Close**.

Figure 4-14

Soon you will receive an email (similar to the one below) containing information about the success (or failure) of the import.

Figure 4-15

As you can see, most of our destinations were uploaded. The ones which the system failed to create are in the attached file.

Figure 4-16

These two destinations were rejected because there was no country code provided, and the system was unable to fill it in automatically, i.e. there was no existing prefix in the system which 259 or 809537 would fall under. So you have to fill in the country code manually, save the file, and repeat the upload. The file should look like the one in Figure 4-15. Note that '—' as the country code has the special meaning "Not Available". This can be used for special destinations (for example, Inmarsat or your company's internal numbering plan) which do not belong to any country.

Figure 4-17

Repeat the destination upload, but upload the **Cannot_Be_Created.csv** file this time.

After you have imported all of the missing destinations in the system, upload again the file with data which was rejected the first time due to missing destinations (**Destination_Unknown**). This time it should import without any problems.

As a last step, verify that the data for your tariff is correct. From the main menu, go to **Tariffs** and then click the ▦ (edit rates) icon to the right of the tariff description. After the rates screen appears, try to look up some rates. Type in a prefix, then click on the **Prefix** button to show all rates for destinations under this prefix. Or, alternatively, you can search by country name (the search is case-insensitive).

Figure 4-18

When is it impossible to import tariff data?

Despite all the flexibility and power of the PortaBilling template engine, sometimes it may still not be possible to directly import certain files. Let's take a look at the most typical cases.

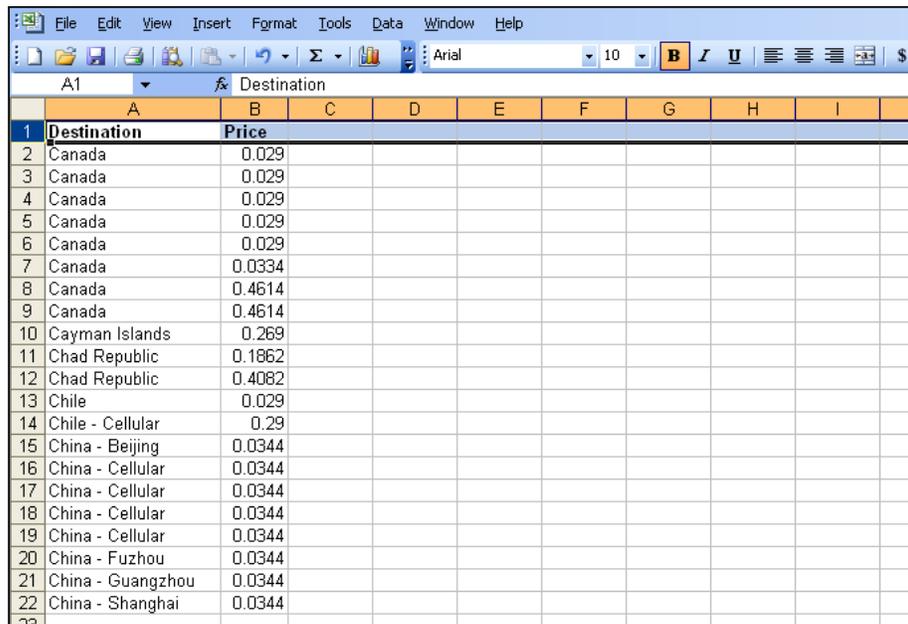# File format is not supported by PortaBilling

It could be that your vendor sends you data in PDF, Microsoft Word, or another format which is not currently supported by the PortaBilling template engine. PortaBilling currently supports:

- Comma-separated values files (CSV)
- Microsoft Excel files (XLS)

So either ask your vendor to use a different format, or try to perform conversion yourself before importing the data.

There is no phone prefix information in the file

Unfortunately, this is a very common situation: your partner sends you a list of rates, but only includes the country or location name, as in the example below.
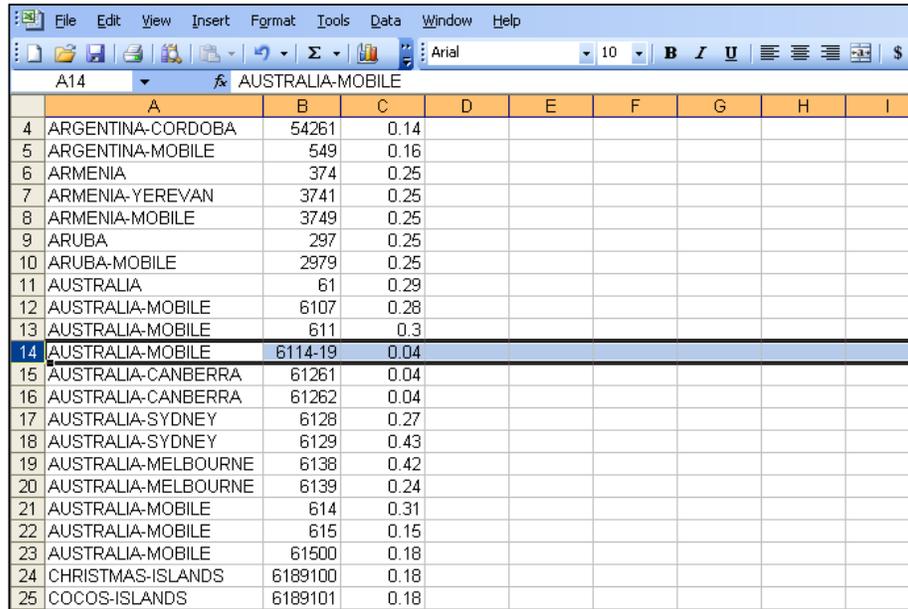

Figure 4-19

It is not feasible to import such data because you can never be sure which destinations were meant under the description "Albania-Cell". Different vendors could use different numberings for that location. The only solution here is to request that your partner provide you with a list of prefixes.
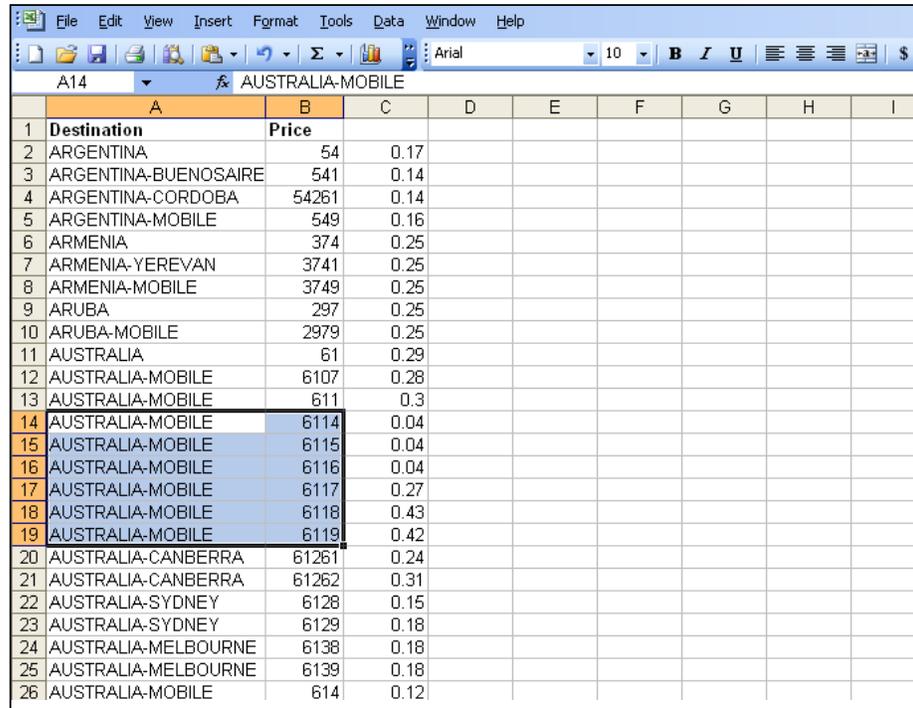
There is more than one phone prefix per data line

Sometimes you are supplied with data using a special coding for prefixes. See the example below.

Figure 4-20

This is also not suitable for uploading. It is necessary to convert this file first (using macros in Excel, or some other tool) so that one line contains a single prefix. Cases like row 37 in the file in Figure 4-20 should be replaced by three lines, as shown below.

Figure 4-21

5. Templates for outgoing email messages

Mail Templates

There are several types of email messages sent from PortaBilling to the end user: notifications about a balance being dangerously close to the credit limit, emails with statistics and/or an invoice, and so on. It is possible to customize the content of these messages, as well as to use templates in different languages.

General template structure

The default mail templates directory is `/home/porta-admin/mail-templates` (on the slave server). You may store templates in a different location; in that case, the `MailTemplatesDir` parameter in the `porta-admin.conf` file should be updated. Hereinafter we shall refer to the directory containing templates as `$mail_templates_dir`.

The default mail templates profile is $default_mail_templates_dir/default. Templates from this location will be used unless a more specific template is found.

For easier understanding and management, mail templates are divided into subdirectories, with each subdirectory containing template files for a separate category of templates. Currently, the following categories are available:

- account
- account_generation
- alarm
- cc_staff
- custom_report
- customer
- destination
- invoice
- payment
- representative
- subscription
- tariff
- user
- vendor

Templates are stored as separate files inside the subdirectory, e.g. the `credit_limit` directory contains the `exceeded.tmpl` and `warning.tmpl` files. These templates are "global", i.e. they apply to all users of your system; see below for a description of templates for a particular language or environment.

Editing templates

You may edit these files directly on the server using any text editor available on the Unix system, or you may copy them to your local workstation (using secure copy, FTP, or another applicable protocol), edit them locally, and then upload them again to the server.

If you plan to make modifications to the templates, it is highly advisable to copy the existing `/home/porta-admin/mail-templates` into a new directory and use it as your new template storage (so that all modifications will be made there). Thus you will have the original versions of the templates intact, and can always revert to them should you encounter any problems with customized templates.

Creating templates for a particular language

PortaBilling can automatically send an email to a customer in his preferred language (according to the Web Interface Language property for the recipient of the email).

The template to be used for a given language should be stored in a subdirectory whose name corresponds to the two-letter language code. For instance, templates for French will reside in the `$mail_template_dir/fr/` subdirectory, templates for Chinese in `$mail_template_dir/ch/`, and so on (see below).

```
$mail_template_dir/default/credit_limit/exceeded.tmpl
$mail_template_dir/fr/credit_limit/exceeded.tmpl
$mail_template_dir/ch/credit_limit/exceeded.tmpl
…
```

When PortaBilling sends a certain email (e.g. notification that a credit limit has been exceeded) it will first check whether there is a template file in the directory for the recipient's language (e.g. `fr/credit_limit/exceeded.html`). If such a file does not exist, then the default template (`default/credit_limit/exceeded.html`) will be used.

Creating templates for a specific environment

If you are providing hosted PortaSwitch services, every environment needs to have its own template, since each hosted customer would like to include his own company name, contact information and such in the email.

This is accomplished by creating yet another level of subdirectories (i.e. env_<i_env>, where i_env is the numeric ID of the environment – you can obtain a list of these IDs via the porta-admin.pl utility) under a specific language, so that the tree looks like this:

```
$mail_template_dir/fr/env_1/
$mail_template_dir/fr/env_2/
…
$mail_template_dir/ch/env_1/
$mail_template_dir/ch/env_2/
…
```

Thus the template file for a "balance exceeded" warning for customers in environment 3 in the French language will reside in:
```
$mail_template_dir/fr/ env_3/credit_limit/exceeded.html
```

Which template will be used?

PortaBilling always tries to find the most specific template, in the following order:
1. For a particular language and environment
2. For a particular environment
3. For a particular language
4. Default template

This may be illustrated by the following example. Let us assume that PortaBilling is to send a "credit exceeded" warning to customer ABC from environment 5, and that this customer has his interface language set to Spanish. PortaBilling will then attempt to use the first available one of the following template files (in the specified order):
1. `$mail_template_dir/sp/env_5/credit_limit/exceeded.html`
2. `$mail_template_dir/default/env_5/credit_limit/exceeded.html`
3. `$mail_template_dir/sp/credit_limit/exceeded.html`
4. `$mail_template_dir/default/credit_limit/exceeded.html`

Mail template syntax

NOTE: Mail templates use basic HTML::Mason syntax; you can find more information on the **HTML::Mason homepage**.

Here is an example of the default `credit_limit/warning.tmpl` template:

```
<%method .subject>
Credit limit warning
</%method>

<%method .body>
Dear <% $greetings %>,

We would like to warn you that the current balance on your
customer account <% $companyname ? $companyname : $name %> is <%
$balance %><% $currency %>.
Your credit limit is <% $credit_limit %><% $currency %> and when
your balance reaches it, your service will be blocked.
This is an auto-generated warning, your current warning
threshold is <% $credit_limit_warning %><% $currency %>.
<%args>
$greetings => ''
$companyname => ''
$name => ''
$balance => ''
$credit_limit => ''
$credit_limit_warning => ''
$currency => ''
</%args>
</%method>

<%method .content_type>
text/plain; charset="utf-8"
</%method>
```

Each template contains **subject, body** and **.content_type** sections. Each section may also contain an **args** section, which defines the parameters received by the template (e.g. current customer's balance), so that they can be used later in the body section.

The **subject** section contains the text to be entered in the subject line of the outgoing email (e.g. Low balance warning).

The **body** section contains the text which will be used as the email body text. Any normal text entered here should appear unchanged in the email. If you want to include the current value of a certain parameter in the text, enclose the parameter variable in `<% %>` as shown below:

```
Your current credit limit is <% $credit_limit %>.
```

In fact, you may include mini-pieces of Perl code in `<% %>`. For instance, the line below includes a conditional statement, and will produce either a company name (if one is defined) or a customer name:

```
<% $companyname ? $companyname : $name %>
```

The **.content_type** section defines the type of the outgoing email message (by default it is plain text). For instance if you wish to use HTML-based mail messages you simply need to change `text/plain` to `text/html`.

Available email templates

The following section provides a brief summary of the available email templates and their use.

Template name	Description
account	
reset_password	Sent when a person uses the "Password recovery" feature on the web self-care interface to reset his password. This message contains the new password.
account_generation	
batch_ok and other	Used to compose email reports about the batch in the account generation process. It is not recommended to change these.
alarms	
CALL_ERR_REP, CC_STAFF_REP, MERCH_ACC_ERRO R, NO_XRATE, PAYMENT_ERR, REPLICATION_ERR O and other	Alerts sent to administrators subscribed to a specific mailing list. **NOTE:** Since most email alerts are triggered in the billing engine, they are configured in the billing engine templates on the master server.
cc_staff	
reset_password	Sent when a person uses the "Password recovery" feature on the web self-care interface to reset his password. This message contains the new password.
customer	
exceeded	Sent when a customer has exceeded his credit limit.
closed	Sent to confirm that a customer's account has been permanently closed.
reset_password	Sent when a person uses the "Password recovery" feature on the web self-care interface to reset his password. This message contains the new password.
send_statistics	Sent when a customer's billing period is closed, to provide him with information about calls during that period. (If the "Full statistics" option has been activated for this customer, an archived CSV file containing all CDRs is attached to the message.)

suspended	Sent when a customer has been suspended because of an overdue invoice.
warning	Sent when a customer's credit limit is over the warning threshold.
invoice	
new_invoice_notify	Sent when a new invoice is created or multiple times **before** the invoice due date to warn the customer that the invoice due date is approaching. Contains a brief summary of the invoice, with the actual invoice attached in PDF form.
overdue_invoice_resend_notify	Sent once or multiple times **after** the invoice due date to warn the customer that the invoice is overdue and service may be suspended soon.
payment	
auto_cc_charge_success	Sent when a credit card on file was successfully charged, contains a confirmation of the credit card transaction.
payment_posted	Notifies a customer that a payment was applied to his account.
periodic_payment_error	Sent to administrators subscribed to the "Payment error" mailing list when execution of periodic payment fails (e.g. credit card is blocked).
representative	
reset_password	Sent when a person uses the "Password recovery" feature on the web self-care interface to reset his password. This message contains the new password.
subscription	
web_subscription	Sent when a potential customer completes the online web signup process.
tariff	
uploaded and other	Used to compose email reports about the tariff upload process. It is not recommended to change these.
user	
reset_password	Sent when a person uses the "Password recovery" feature on the web self-care interface to reset his password. This message contains the new password.
vendor	

reset_password	Sent when a person uses the "Password recovery" feature on the web self-care interface to reset his password. This message contains the new password.