

Cacti ISP Billing Documentation

Version 1.0.13

Author: Tony Roman
Copyright © 2012 Cacti Group



Contents

- 1. *Preface***
- 2. *Features***
- 3. *Operations***
 - i. *General***
 - ii. *Output***
 - iii. *Data Collection***
 - iv. *RRA Definitions***
- 4. *Installation***
- 5. *Upgrading***
- 6. *Configuration***
 - i. *Default / Global***
 - ii. *Customers and Graphs***
 - iii. *Billing Time Frame (Interval)***
 - iv. *Rate***
 - v. *Email***
 - vi. *Currency***
 - vii. *Branding Email Output***
 - viii. *Exporting to CSV (Comma Separated Values) File***
 - ix. *Threshold Tracking***
 - x. *Using the build Command***
 - xi. *Example Configuration***
 - xii. *Track File***
- 7. *Testing***
- 8. *CSV File Formats***
 - i. *Email CSV File***
 - ii. *Export CSV File***
- 9. *Command Line Reference***

Revision History

- 2006-11-11 – Document creation*
- 2006-12-20 – Minor changes for grammar and syntax*
- 2007-02-17 – Updates for version 1.0.1, including extended email support*
- 2007-02-24 – Update for version 1.0.2, including additional sections on Upgrading, RRA Definitions, Data Collection and Testing.*
- 2007-05-05 – Update for version 1.0.3 and information on supported Cacti versions.*
- 2007-07-16 – Updated for version 1.0.4*
- 2007-09-24 – Updated for version 1.0.5 and Cacti 0.8.7*
- 2007-12-10 – Updated for version 1.0.6*
- 2008-01-21 – Updated for version 1.0.7*
- 2008-02-03 – Updated for version 1.0.8*
- 2008-11-23 – Updated for version 1.0.9, information about: Features, Rates, Tracking File, etc.*
- 2009-03-08 – Update for version 1.0.10, Threshold tracking added*
- 2010-08-27 – Update for version 1.0.11*
- 2011-05-08 – Update for version 1.0.12, clarification on Bandwidth Summation Thresholds*
- 2012-05-30 – Update for version 1.0.13 and Cacti 0.8.8*

Preface

Section 1

If you are reading this, you have already taken the first steps of using Cacti or thinking about using Cacti to graph your and your customers' bandwidth usage. The ISP billing script is a simple to use command line php script that emails periodic reports of customers' bandwidth usage and calculated owed amounts.

This document assumes you have Cacti up and running and graphing bandwidth usage. It is also assumed that you have at least a basic understanding of Cacti and how to view the graphs.

The system supports multiple types of billing intervals: Yearly, Monthly, Bi-monthly, Weekly and Daily. All these are configurable to allow for maximum flexibility.

Billing rates are also very flexible. Regular billing rates, fixed or committed/overage billing rates can be applied globally, to individual customers or to a single graph.

Minimum Cacti version 0.8.6h is required for the ISP Billing Script. This requirement is present because the 95th Percentile functions were updated extensively in this version of Cacti. The ISP billing script is dependent on Cacti, so version requirements of Cacti apply to the ISP billing script.

If you have any questions about the operation of the script, have a bug to report or would like to request a feature, please contact Tony Roman at roman@disorder.com.

Features

Section 2

The Cacti ISP Billing Script is a comprehensive addon for Cacti that allows for daily, weekly, monthly or bi-monthly per customer billing of Cacti generated graphs for Nth Percentile and Bandwidth Summation.

Multiple or single graphs can be associated with a customer to allow for multiple interface billing for a single customer.

Standard, Committed/Overage and Fixed billing rates are supported per customer.

Email output can be generated per customer. Email output is HTML with graph images and an attached CSV file containing the same data that the email contains. All these options are configurable.

Exported CSV data can be create per customer and post-processed by a user defined script to any format for importing into a billing system. Example scripts are supplied for creation of tab delimited files, SQL insert statements, XML output, etc.

Only Nth Percentile and Bandwidth Summation comment lines on Cacti graphs are considered a billable item. So, the value that a customer will see on the graph will be that is used to calculate billable totals.

Inter-billing period notifications of exceeded committed rate, configurable per customer.

Operations

Section 3

General

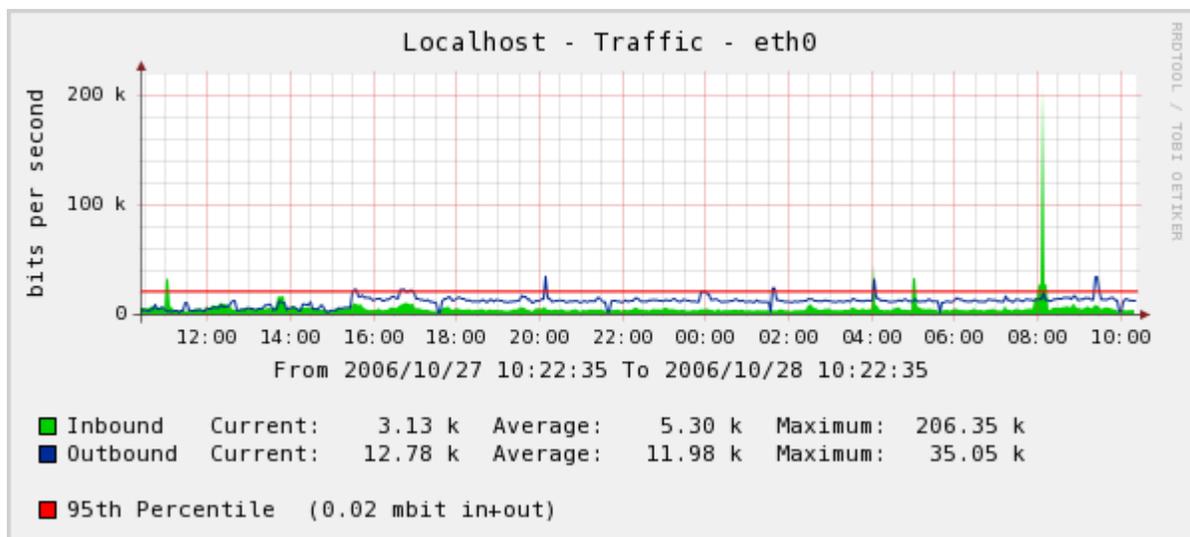
Sub Section i

The ISP billing script is to be run everyday on the server which Cacti is installed. Simply use cron or windows scheduler to schedule the execution of the script. The ISP billing script requires a configuration and a track file (auto generated). The configuration file is where the customers' information is defined. The track file is where the last time billing was performed for the configured customers is saved. Multiple configuration files can be used for different groups of customers, all that needs to be done is schedule each configuration separately to be run daily. Multiple configurations can use the same track file, but it is suggested that separate track files are used per configuration file, as duplicate customer descriptions between configuration files can collide in the track file.

What does the ISP billing script use from Cacti to determine what is billable?

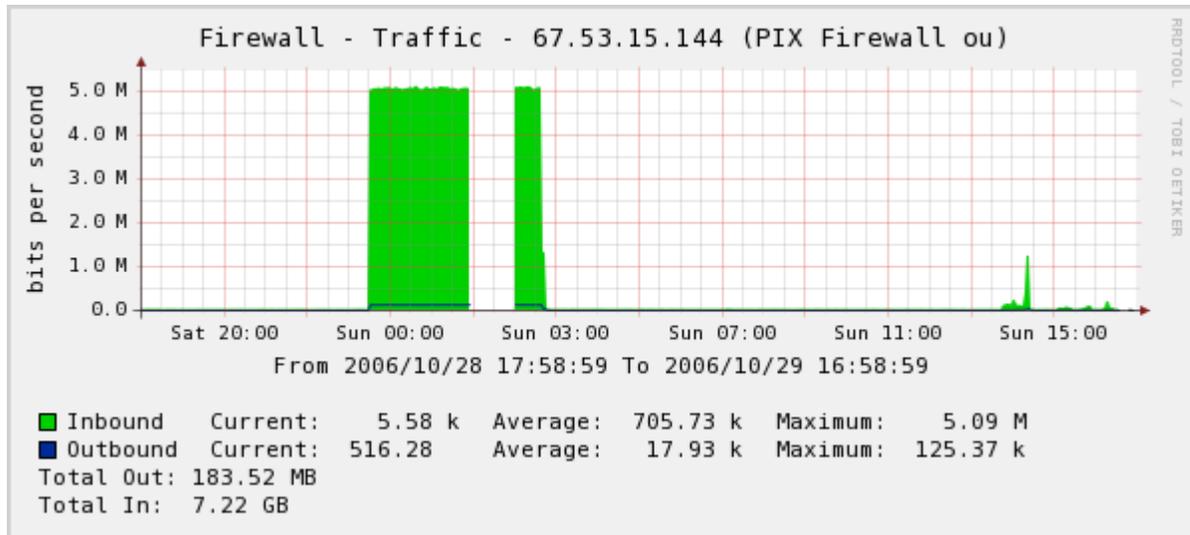
The ISP billing script only looks at graphs with graphs items types of comments that contain graph variables that defined Bandwidth Summation and Nth Percentile values.

Example 1 – Nth Percentile Graph



The resulting billable value would be “95th Percentile (0.02 mbit in+out)”.

Example 2 – Bandwidth Summation Graph



The resulting billable value would be “Total Out: 183.52 MB” and “Total In: 7.22 GB”.

Billing Intervals

There are 4 available billing intervals:

Daily:

Billing occurs daily or every nth number of days defined from midnight to midnight.

Weekly:

Billing occurs weekly or every nth weeks on Saturday night at midnight to Saturday night at midnight.

Monthly:

Billing occurs monthly on the defined day, 1 to last day of month, every 1 or nth months on midnight of the defined day.

Bi-monthly:

Billing occurs monthly on the 15th and last day of the month.

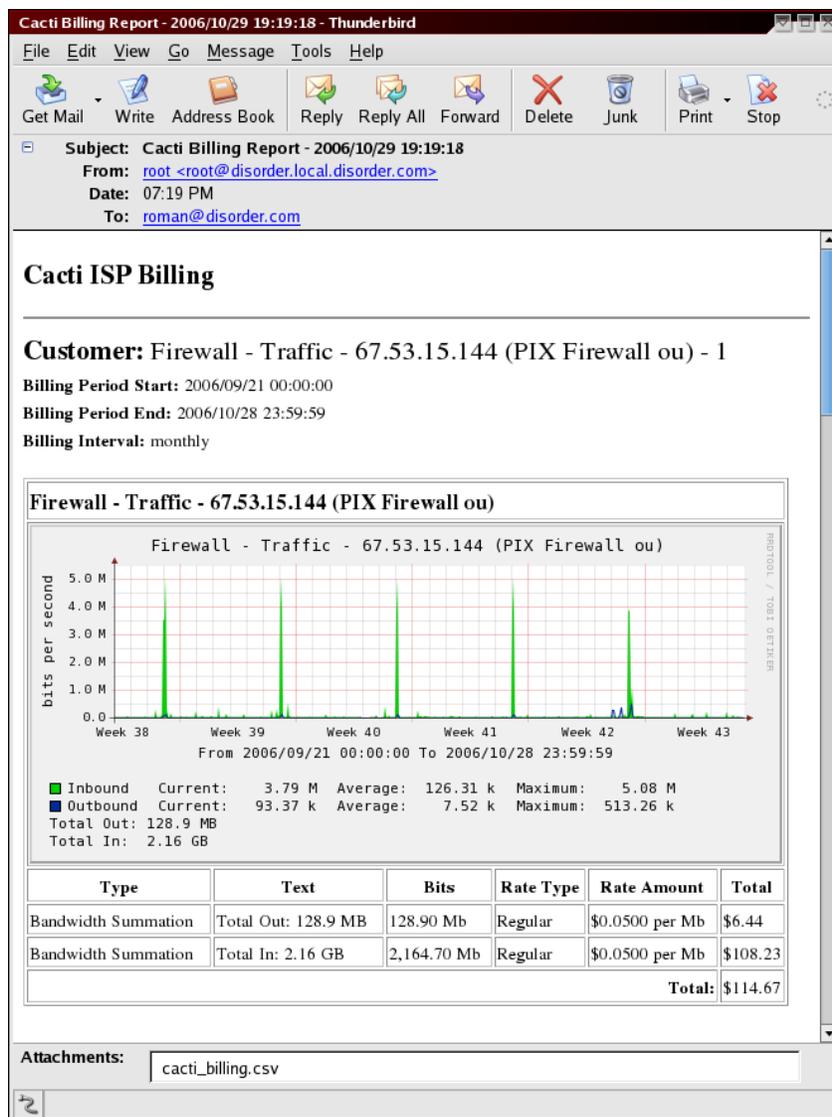
Note: All billing intervals are processed on the day after the ending day of the interval. For example, monthly for October would be from 10-1 00:00:00 to 10-31 23:59:59, but wouldn't be processed until the billing script is run on 11-1.

Output

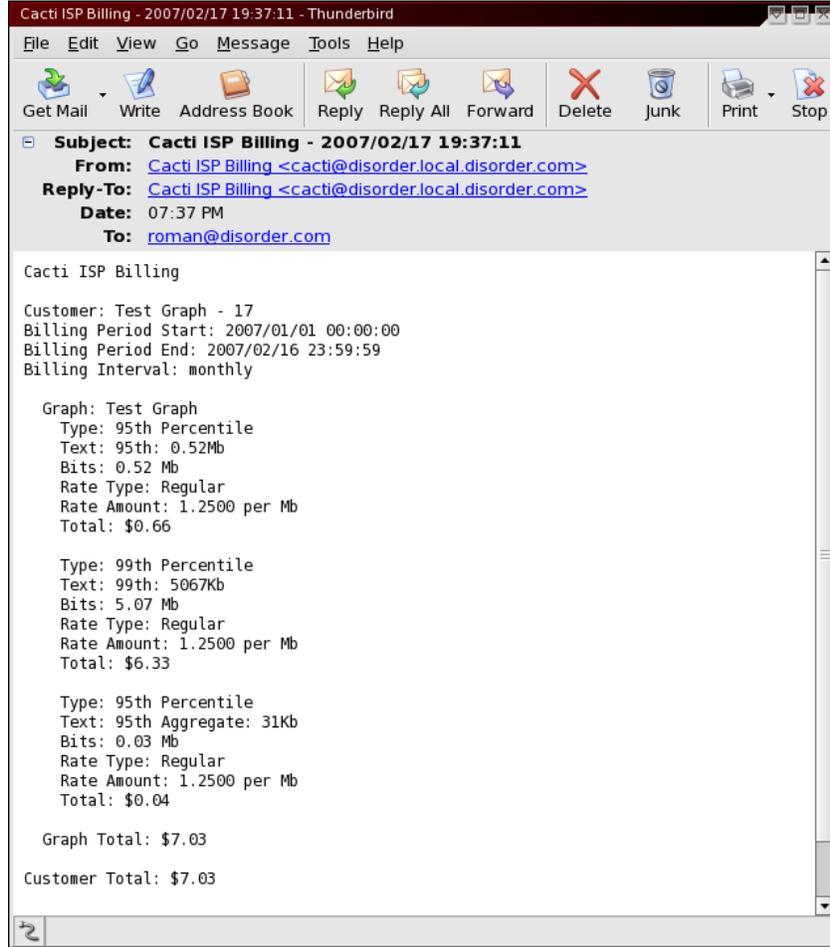
Sub Section ii

The ISP Billing script supports multiple output types within the emailed reports. HTML and text formatted messages are enabled by default, but are can be enabled or disabled per defined email address.

Example 3 – HTML email billing report



Example 4 – Text only billing report



Emails by default include an attached CSV file that mime type is set to open with Excel or Open Office Spreadsheet.

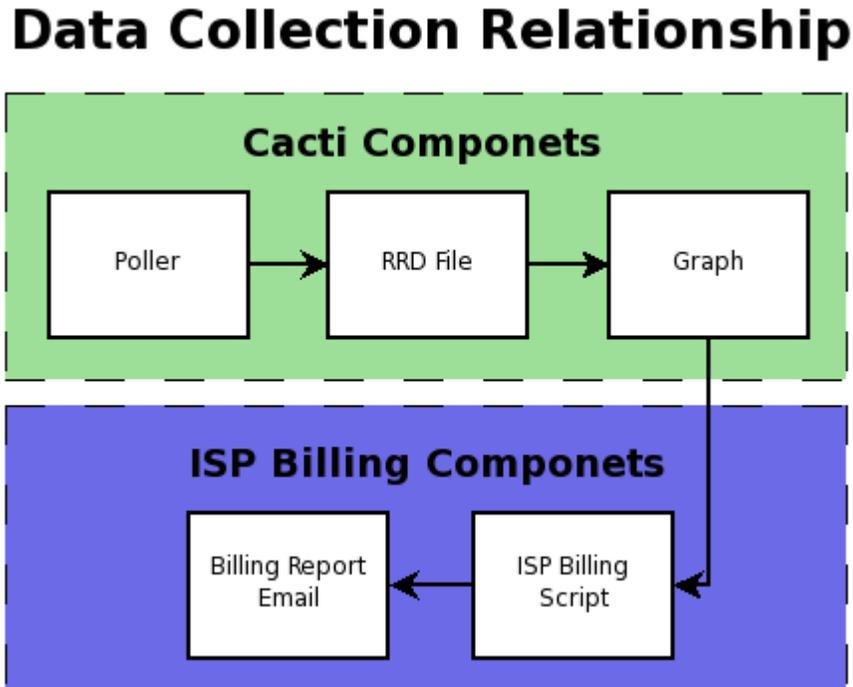
Emails default to HTML format, per email address parameters allow for text only emails and removal of the CSV attachment.

Data Collection

Sub Section iii

Data collection seems to be a subject that comes up a lot in conjunction with the ISP Billing script. The ISP Billing script uses the graph definitions defined in Cacti. However the data is collected, the ISP Billing script has no concept of this, it relies on Cacti to perform this action.

Example 5 – Data Collection Relationship



It is often asked what data the ISP Billing script will use from the RRDTool files. The answer is simple, it will let RRDTool figure it out. Now, what does RRDTool do, it attempts to use the highest resolution data set that contains data for the selected time frame. This is why it is very important, for billing purposes, that you retain six months to a year of high resolution data. A great example would be to retain one year of 5 Minute Average (Daily) information. Please refer the the next section for information on defining and updating your RRA definitions in Cacti to retain the recommended higher resolution data.

RRA Definitions

Sub Section iv

Most people either don't understand or don't realize that Cacti uses RRDTool to store the information it gathers from devices. The MySQL database is simply the place that Cacti stores the configuration information on what it will poll, where it will store it and how to present the stored data. RRDTool files are where the statistical data is stored. The beauty of RRDTool files is that once they are created they do not grow or shrink in size, this is the nature of a Round Robin Database.

For more information about RRDTool, please visit the RRDTool website at: <http://www.rrdtool.org>

Because you are using the Cacti ISP Billing script, we can only assume that you are using Cacti for billing purposes. This being said, it's very important that your RRDTool files retain at least three times the maximum billing interval of high resolution data (5 Minute Average by Default in Cacti).

Examples:

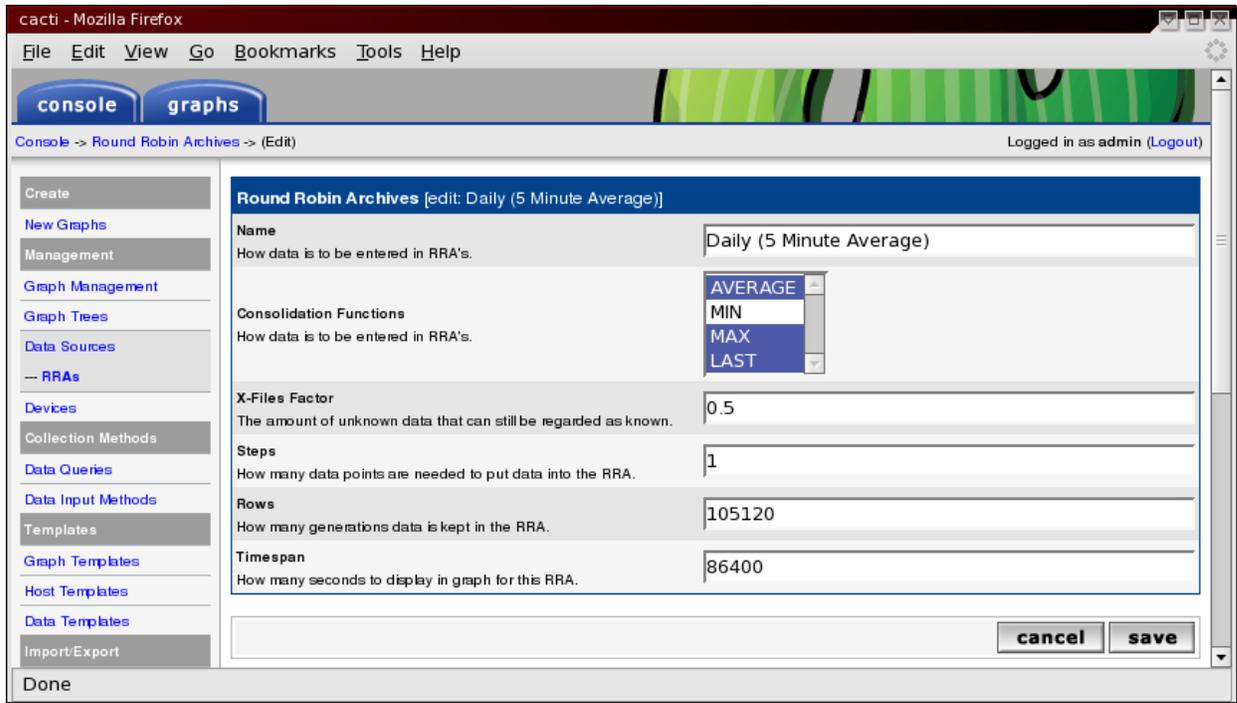
1. If you had a maximum billing interval of every 3 months, you would want to retain at least nine months of high resolution data.
2. If you have a maximum billing interval of monthly, you would want to retain at least three months of high resolution data.

High resolution data is considered to be which ever is your lowest consolidation function defined in Cacti's RRA settings the default being "Daily (5 Minute Average)".

Example 6 – Cacti RRA (Round Robin Archives)

Name	Steps**	Rows	Timespan	Add
Daily (5 Minute Average)	1	105120	86400	✗
Weekly (30 Minute Average)	6	8736	604800	✗
Monthly (2 Hour Average)	24	2160	2678400	✗
Yearly (1 Day Average)	288	797	33053184	✗

Example 7 – Daily (5 Minute Average) one year high resolution data



The above examples have been altered to retain one year of Daily (5 Minute Average) high resolution data.

WARNING: Defining large high resolution RRA definitions can degrade system performance and use a considerable amount of disk space. One traffic interface with the default two day high resolution retention will result in a 95K sized RRDTool file, but the same file with one year of high resolution retention will result in a 9M file. Multiple that change by all the RRDTool file that Cacti uses and it can be a considerable amount of disk space. PHP execution time and memory maximums settings will have to be increased to accommodate the time and resources needed to process such data sets.

Daily (5 Minute Average) RRA Retention length	Rows
2 Days (Cacti Default)	600
3 Months	26,280
6 Months	52,560
9 Months	78,840
1 Year	105,120

Updating your existing RRDTOol files

One thing Cacti will not do for you is alter RRDTOol files that have already been created. You have to “rrdtool resize” the files to have the correct number of defined rows.

Please refer to the rrdtool documentation on instructions for doing this.

There is a very helpful post on the Cacti forums on how to do this and a script to assist in doing so.

<http://forums.cacti.net/viewtopic.php?t=11425>

Installation

Section 4

1. Unpacking the program

The file is supplied in a tar.gz format. Use the following command to extract:

```
gunzip -c isp_billing*.tar.gz | tar -xvf -
```

If you are on Windows, you can use your favorite archiving program that supports tar.gz. Winzip is a great example of an archive program that will extract these files.

2. Once the isp_billing directory is extracted, it needs to be moved to the directory that Cacti is installed. Example command:

```
mv isp_billing /var/www/htdocs/cacti/
```

3. To ensure you don't get any pesky PHP errors about sessions, it's important to add the isp_billing.php to the \$no_http_header_files array in the include/config.php (0.8.6j-) or include/global.php (0.8.7+) file for Cacti. If you are running windows, this change may not be needed.

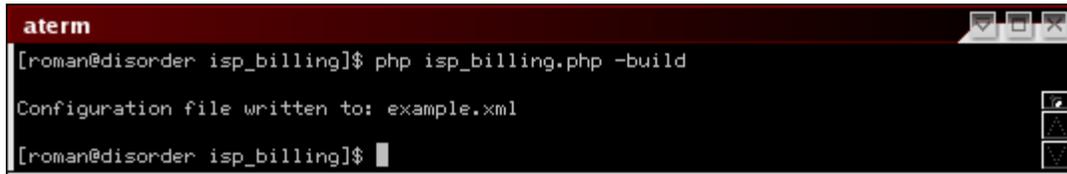
```
/* Files that do not need http header information - Command line scripts */
$no_http_header_files = array(
    "poller.php",
    "cmd.php",
    "query_host_cpu.php",
    "query_host_partitions.php",
    "sql.php",
    "ss_host_cpu.php",
    "ss_host_disk.php",
    "ss_sql.php",
    "isp_billing.php"
);
```

4. Before you can add the scheduled jobs for the billing script a configuration file has to be generated. The isp_billing script has a built in function to build a configuration off of your already created graphs in Cacti. As mentioned earlier, it will only uses graphs that contain comment fields with graph variables defined for Nth Percentile and Bandwidth Summation. The build process also makes some assumptions, like a bill rate of \$0 and monthly billing interval processed on the last day of the month. For more information on how to configure these items and usage of the build command, please refer to Chapter 4 on Configuration.

Use the following command to build an example.xml configuration file:

```
php isp_billing.php -build
```

Example 8 – Output of build command



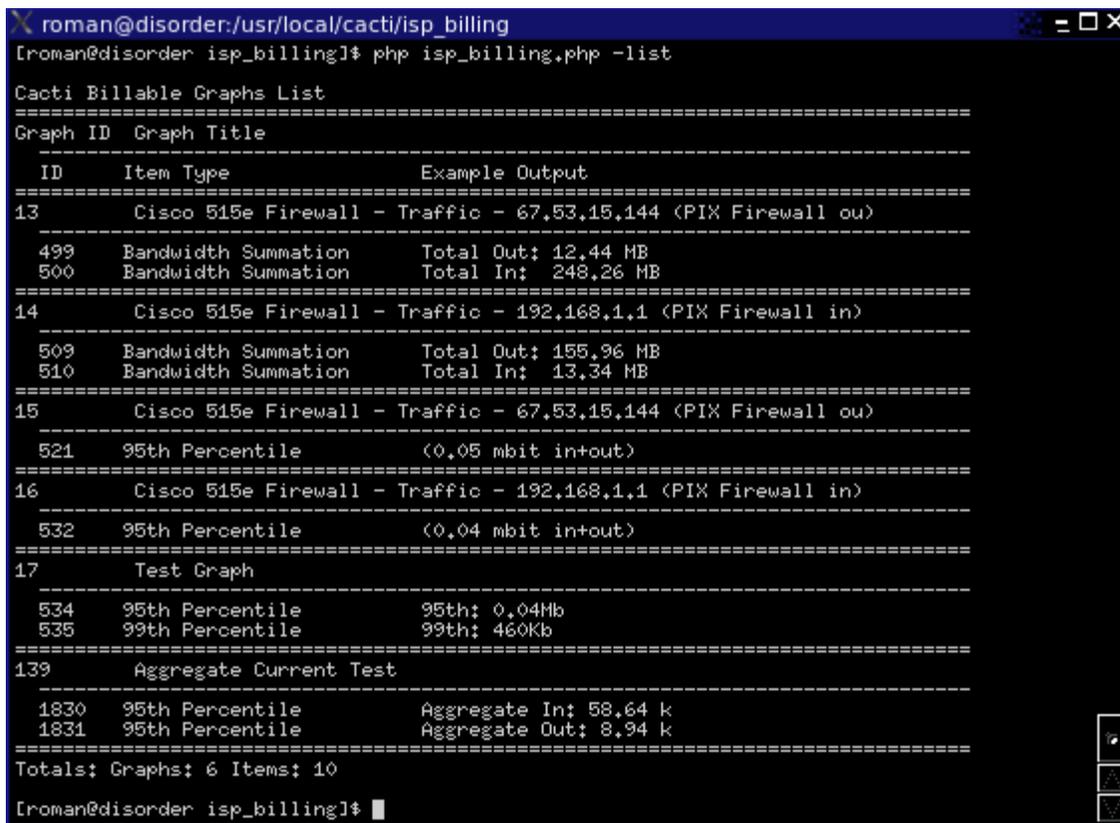
```
aterm
[roman@disorder isp_billing]$ php isp_billing.php -build
Configuration file written to: example.xml
[roman@disorder isp_billing]$
```

Using the list function can generate a more readable summary format. Use the following command line to execute the command:

```
php isp_billing.php -list
```

The resulting output will aid you in determining which graphs to configure for what customers.

Example 9 – Output of list command

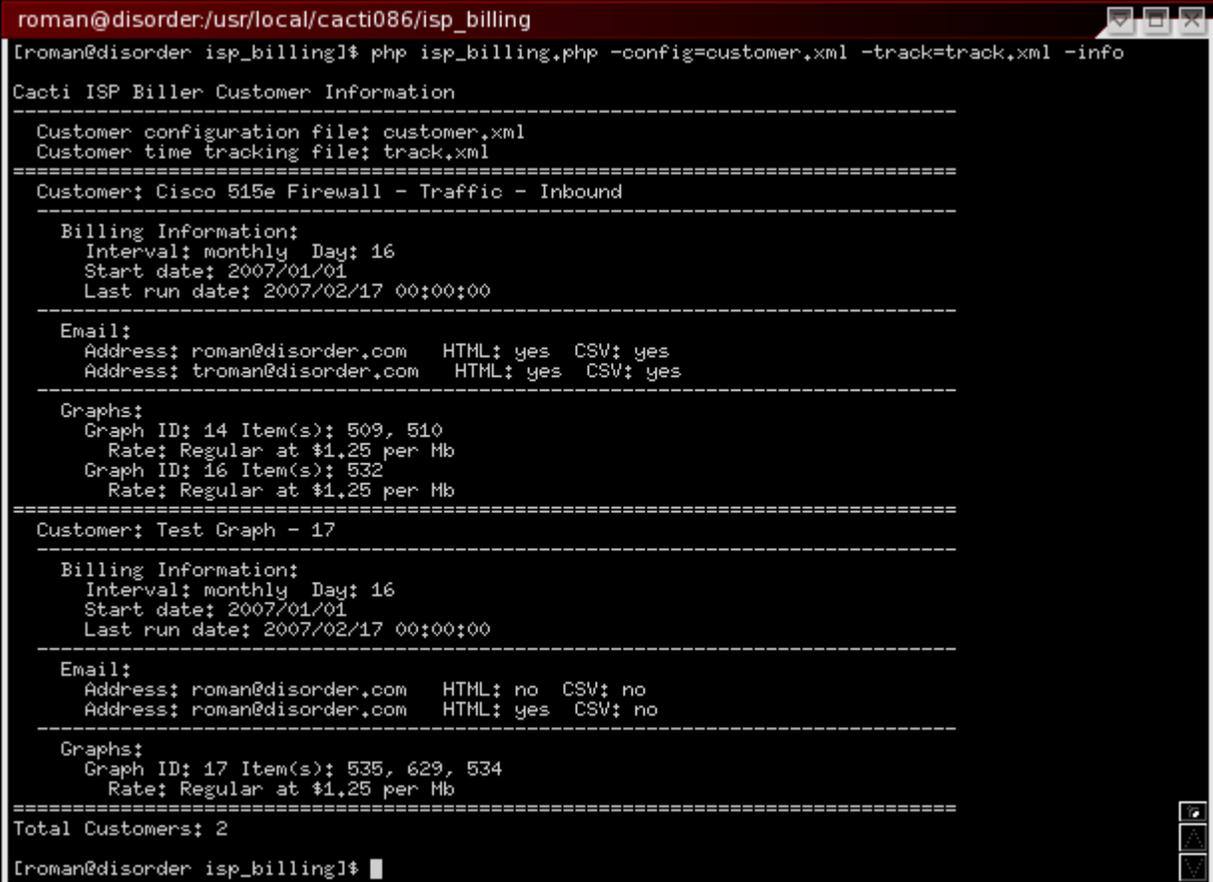


```
roman@disorder:/usr/local/cacti/isp_billing
[roman@disorder isp_billing]$ php isp_billing.php -list
Cacti Billable Graphs List
=====
Graph ID  Graph Title
=====
  ID      Item Type          Example Output
=====
  13      Cisco 515e Firewall - Traffic - 67,53,15,144 (PIX Firewall ou)
=====
  499     Bandwidth Summation    Total Out: 12,44 MB
  500     Bandwidth Summation    Total In: 248,26 MB
=====
  14      Cisco 515e Firewall - Traffic - 192,168,1,1 (PIX Firewall in)
=====
  509     Bandwidth Summation    Total Out: 155,96 MB
  510     Bandwidth Summation    Total In: 13,34 MB
=====
  15      Cisco 515e Firewall - Traffic - 67,53,15,144 (PIX Firewall ou)
=====
  521     95th Percentile        (0,05 mbit in+out)
=====
  16      Cisco 515e Firewall - Traffic - 192,168,1,1 (PIX Firewall in)
=====
  532     95th Percentile        (0,04 mbit in+out)
=====
  17      Test Graph
=====
  534     95th Percentile        95th: 0,04Mb
  535     99th Percentile        99th: 460Kb
=====
  139     Aggregate Current Test
=====
  1830    95th Percentile        Aggregate In: 58,64 k
  1831    95th Percentile        Aggregate Out: 8,94 k
=====
Totals: Graphs: 6 Items: 10
[roman@disorder isp_billing]$
```

Once you have built your configuration file, you can check to see if it is what you want by using the info command line option to review the configuration. The info command line option is very useful, it will allow you to get information about configured customers in each configuration file you have. To use the info function, use the following command line:

```
php isp_billing.php -config=customer.xml track=track.xml -info
```

Example 10 – Output of info command



```
roman@disorder:/usr/local/cacti086/isp_billing
[roman@disorder isp_billing]# php isp_billing.php -config=customer.xml -track=track.xml -info
Cacti ISP Biller Customer Information
-----
Customer configuration file: customer.xml
Customer time tracking file: track.xml
-----
Customer: Cisco 515e Firewall - Traffic - Inbound
-----
Billing Information:
Interval: monthly Day: 16
Start date: 2007/01/01
Last run date: 2007/02/17 00:00:00
-----
Email:
Address: roman@disorder.com HTML: yes CSV: yes
Address: troman@disorder.com HTML: yes CSV: yes
-----
Graphs:
Graph ID: 14 Item(s): 509, 510
Rate: Regular at $1,25 per Mb
Graph ID: 16 Item(s): 532
Rate: Regular at $1,25 per Mb
-----
Customer: Test Graph - 17
-----
Billing Information:
Interval: monthly Day: 16
Start date: 2007/01/01
Last run date: 2007/02/17 00:00:00
-----
Email:
Address: roman@disorder.com HTML: no CSV: no
Address: roman@disorder.com HTML: yes CSV: no
-----
Graphs:
Graph ID: 17 Item(s): 535, 629, 534
Rate: Regular at $1,25 per Mb
-----
Total Customers: 2
[roman@disorder isp_billing]#
```

5. Scheduling the isp_billing script:

Unix uses cron for this purpose, this script can be run as any user, but should not be run as root. The only consideration to understand is the user running the script must have read access to the rrd files in the Cacti installation. Use the following line in your global crontab to schedule the isp_billing script. Remember to input the full path to php and use the correct user account.

```
0 1 * * * cactiuser php /var/www/htdocs/cacti/isp_billing/isp_billing.php \  
-config=/var/www/htdocs/cacti/isp_billing/config/customers.xml \  
-track=/var/www/htdocs/cacti/isp_billing/config/track.xml
```

It is important to note that you need to replace the paths with the correct paths for your system.

Upgrading

Section 5

Because you might already have a previous version of the ISP Billing Script, upgrading must be covered.

Upgrading is easy!

1. Move your current ISP Billing directory to a backup location.

```
mv /var/www/html/cacti/isp_billing /var/www/html/cacti/isp_billing.backup
```

2. Extract and move or copy the new ISP Billing directory to your Cacti installation.

```
mv /tmp/isp_billing /var/www/html/cacti/isp_billing
```

3. Copy your configuration files from the backup to the new installation.

```
cp -p /var/www/html/cacti/isp_billing.backup/config/* /var/www/html/cacti/isp_billing/config
```

4. Perform a test run to confirm that everything is good.

Note: Above are just example paths. Please use the proper paths for your system.

Configuration

Section 6

The configuration file for the ISP billing script is a specifically formatted XML file. In the following sections, the format and parameters of the configuration sections will be reviewed and explained.

The configuration file must begin with the following:

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>  
<cacti_bill>
```

and end with the following:

```
</cacti_bill>
```

Note: “UTF-8” encoding should be used on anything other than Windows. On Windows, use “ISO-8859-1” encoding.

Once you have your configuration ready, you should always check it. Use the following command line option to check the configuration for errors:

```
php isp_billing.php -check -config=customer.xml -track=track.xml
```

Default / Global

Sub Section i

Default / Global section is required. Values that are not defined per customer or per graph then use the values defined in this section of the configuration file.

Example of defaults / globals section:

```
<global>
<defaults>
  <email>billing@isp.com</email>
  <rate unit="Mb">0.05</rate>
  <billing_timeframe>
    <type>monthly</type>
    <day>last</day>
    <every>1</every>
    <start_date>2006-10-01</start_date>
  </billing_timeframe>
  <currency>
    <pre>$</pre>
    <post> USD</post>
    <precision>2</precision>
  </currency>
  <file process="yes">export.csv</file>
</defaults>
<email>
  <return_address>billing@isp.com</return_address>
  <subject>Bandwidth Billing</subject>
  <title>Bandwidth Billing</title>
  <csv_file_name>billing</csv_file_name>
</email>
<export_file_processor>scripts/post_process.pl</export_file_processor>
<threshold>
  <enabled>no</enabled>
  <notification>no</notification>
  <subject>Notification Email Subject</subject>
  <title>Email Address Title</title>
</threshold>
</global>
```

Valid parameters for global section email settings are covered later in this section.

Valid parameters for the rate and billing_timeframe sections are covered later in this section.

Valid parameters for currency section are covered later in this section.

Valid parameters for export_file_processor and file are covered later in this section.

Valid parameters for threshold are covered later in this section.

An email address must be defined in the default section, email address should also be defined per customer. Any customer that does not have an email defined will use the defined default email. Refer

to the email section for optional parameters.

Customers and Graphs

Sub Section ii

At least one customer section is required for a working configuration.

Customer Fields:

XML Field	Description
<customers>	Customers container opening tag
<customer>	Customer container opening tag
<description></description>	This is the customer description, this value can not be duplicated between customers. This value will appear on the billing reports as the customers description.
<file></file>	Optional field – Refer to the CSV export section for parameters and values.
<external_id></external_id>	Optional field – Only used by CSV export for potentially identifying customer in external system.
<rate></rate>	Optional field – Refer to rates section for parameters and values.
<billing_timeframe></billing_timeframe>	Optional field – Refer to billing_timeframe section for parameters and values.
<email></email>	Optional field – Email address to email the resulting report to. Multiple entries are supported. To disable emailing use the string value of “NULL”.
<currency></currency>	Optional field – Refer to currency section for parameters and values.
<graphs>	Graphs container tag, required.
<graph>	At least one graph section must be defined, multiple are allowed.
<rate></rate>	Optional field – Refer to rates section for parameters and values.
<external_id></external_id>	Optional field – Only used by CSV export for potentially identifying customers' graph in external system.
<id></id>	Graph id – This value can be gotten from the Cacti web interface by viewing the graph in question and examining the URL for the variable called “local_graph_id”.
<graph_item></graph_item>	Optional field – Define the exact graph item on the graph to process billing. If no graph_items are defined, any matching comment fields will be used from the defined graph.
</graph>	Graph container closing tag

XML Field	Description
</graphs>	Graphs container closing tag
</customer>	Customer container closing tag
</customers>	Customers container closing tag

Note: Altering the customer description after you have processed billing will require the same modification to the track file description.

Example Customer XML:

```

<customers>
  <customer>
    <description>Firewall - Traffic - 192.168.1.1 (PIX Firewall in) – 1</description>
    <email>billing@isp.com</email>
    <email>customer@isp.com</email>
    <rate unit="Mb">1.00</rate>
    <billing_timeframe>
      <type>monthly</type>
      <day>28</day>
      <every>1</every>
      <start_date>2006-09-21</start_date>
    </billing_timeframe>
    <graphs>
      <graph>
        <id>14</id><!-- Graph Title: "Firewall - Traffic - 192.168.1.1 (PIX Firewall in)" -->
        <graph_item_id>509</graph_item_id><!-- Comment: "Total Out: |sum:auto:current:2:auto|"-->
        <graph_item_id>510</graph_item_id><!-- Comment: "Total In: |sum:auto:current:2:auto|"-->
        <rate unit="Mb" type="committed">0.50</rate>
        <rate unit="Mb" type="overage" threshold="1000000000">2.00</rate>
      </graph>
      <graph>
        <id>16</id><!-- Graph Title: "Firewall - Traffic - 192.168.1.1 (PIX Firewall in)" -->
      </graph>
    </graphs>
  </customer>
</customers>

```

In the example above, the customer is defined with committed/overage rate on the graph with an id of 14 and the graph with the id of 16 will use the customer defined rate.

Billing Time Frame (Interval)

Sub Section iii

Billing time frame can be defined globally or per customer. If not supplied at a customer level, then the global default is used.

Customer Fields:

XML Field	Description
<billing_timeframe>	Billing time frame container opening tag.
<type></type>	Defined billing time frame type: daily, weekly, monthly or bi-monthly. daily – billing occurs daily or every nth days as defined by the every field. weekly – billing occurs weekly or every nth weeks as defined by the every field. monthly – billing occurs monthly on the defined day, 1 to last day of the month, every nth months as defined by every. bi-monthly – billing occurs twice monthly on the 15 th and last day of the month.
<day></day>	Used only when type of monthly is defined and denotes the day of the month with billing reoccurs. Using the keyword 'last' denotes to use the last day of the month being billed.
<every></every>	Used by daily, weekly and monthly types, this denotes every nth days, weeks or months to bill. Example, you want to bill every 3 months, you would define every with a value of 3.
<start_date></start_date>	Optional field to define when to start billing a customer. This allows for you to pre-configure a customer and not get billing reports until needed.
</billing_timeframe>	Billing time frame container closing tag

Example Billing Time Frame XML Sections:

```
<billing_timeframe>  
  <type>daily</type>  
  <every>1</every>  
  <start_date>2006-09-21</start_date>  
</billing_timeframe>
```

```
<billing_timeframe>  
  <type>weekly</type>  
  <every>2</every>  
  <start_date>2006-09-21</start_date>  
</billing_timeframe>
```

```
<billing_timeframe>  
  <type>monthly</type>  
  <day>28</day>  
  <every>1</every>  
  <start_date>2006-09-21</start_date>  
</billing_timeframe>
```

```
<billing_timeframe>  
  <type>bi-monthly</type>  
  <start_date>2006-09-21</start_date>  
</billing_timeframe>
```

Rate

Sub Section iv

Billing rate section can be defined globally, per customer or per graph. There are three types of billing rates Regular, Fixed and Committed/Overage.

Committed/Overage rate items must be defined together. Regular, Fixed and Committed/Overage can not be defined on the same global, customer or graph. If this done, the Regular rate will be applied.

Rate defined parameters:

XML Field	Description
type	Type of billing rate, undefined (regular), fixed, committed or overage. Committed and overage must be defined together.
unit	Units to bill on, Kb (1000 bits), Mb (1000000 bits) or Gb (1000000000 bits).
threshold	Only defined when type is overage, this is the number of bits or bytes , allowed before the overage rate is applied. Overage rates are only applied to the amount over this defined number. Thresholds for Nth Percentile are expressed in bits, but thresholds for Bandwidth Summation are defined in bytes.
minimum	Only defined when type is “committed” or “undefined” (Regular). This is the minimum amount the customer will be billed. If the calculated rate falls below this amount, the minimum will be the applied. When applied to “committed” rate and the rate amount is set to “0”, the rate will be displayed as fixed.

Example Rate Regular (Undefined) XML Section:

```
<rate unit="Mb">0.50</rate>
```

Example Rate Committed/Overage XML Section:

```
<rate unit="Gb" type="committed">0.25</rate>  
<rate unit="Gb" type="overage" threshold="100000000">0.50</rate>
```

Example Rate Committed/Overage with fixed Committed Rate XML Section:

```
<rate unit="Gb" type="committed" minimum="100">0</rate>  
<rate unit="Gb" type="overage" threshold="100000000">0.50</rate>
```

Example Rate Fixed XML Section:

```
<rate type="fixed">100.00</rate>
```


Email

Sub Section v

Email entry can be defined globally or per customer. There are two options that can be configured per email address: HTML formatted report and if the CSV file is attached to the report.

Multiple email entries can be defined, either globally or per customer. Reports are emailed out in a consolidated fashion based on the configured options for the email addresses. Example, if you have the email address `billing@isp.com` configured with `html="no"` on all but one customer, all but that one customer will result in a text only email to `billing@isp.com`, but the one email configured with `html="yes"` (default) will generate a separate email to `billing@isp.com`.

To disable emailing use the string value of "NULL".

Email defined parameters:

XML Field	Description
html	"yes" or "no", default "yes", defined if email report are HTML formatted with images.
csv	"yes" or "no", default "yes", defined if email report has CSV attachment.
type	"all", "billing" or "threshold" - Determines what type of emails this address will get. all: Billing and Threshold notification emails billing: Billing email reports only threshold: Threshold notification emails only

Example Email XML Parameters:

```
<email html="yes" csv="yes">customer@isp.com</email>
<email html="yes" csv="yes" type="all">customer@isp.com</email>
<email html="yes" csv="yes" type="billing">customer@isp.com</email>
<email html="no" csv="yes">billing@isp.com</email>
<email>accounting@isp.com</email>
<email type="threshold">tech@isp.com</email>
<email>NULL</email>
```

Currency

Sub Section vi

The currency section of the configuration can be defined globally in the defaults section and/or per customer. This allows for per customer calculations in difference currencies and/or defining default currency to use for all customers.

Currency defined section:

XML Field	Description
<currency>	Currency container opening tag.
<pre></pre>	Pre text inserted before amounts, default: "\$"
<post></post>	Post text inserted after amounts, default: ""
<precision> </precision>	Number of decimal places to use for amount calculations, default: "2"
</currency>	Currency container closing tag.

Example Rate Regular XML Section:

```
<currency>  
  <pre>$</pre>  
  <post> USD</post>  
  <precision>2</precision>  
</currency>
```

Branding Email Output

Sub Section vii

Branding is important, even if you don't plan on emailing the reports to your customers, it is important to brand the emails. This is done through some global configuration parameters that are defined in the configuration file. This is only defined in the global section of the configuration, not on a customer basis.

Global Email fields:

XML Field	Description
<global>	Globals opening tag
<email>	Email opening tag
<return_address> </return_address>	Email address the billing reports appear to be coming from.
<subject></subject>	Email subject
<title></title>	Email title, also used as the name of the return email address.
<csv_file_name> </csv_file_name>	This is the name of the attached csv file without the “.csv” extension.
<footer></footer>	Footer text to add to the end of billing email reports
</email>	Email closing tag
</globals>	Globals closing tag

Example Globals Email XML Section:

```
<global>
<email>
  <return_address>return@isp.com</return_address>
  <subject>My ISP Billing</subject>
  <title>My Billing Title</title>
  <csv_file_name>my_isp_billing</csv_file_name>
  <footer>This is considered your billing invoice.</footer>
</email>
</global>
```

Note: Do not add additional “<global></global>” tags, simply place the above section in the existing global section of your configuration file.

Exporting to CSV (Comma Separated Values) File

Sub Section viii

The ISP Billing script supports defining a global or a per customer export file using the “file” parameter. The export files can be post processed by an external script by defining the “export_file_processor” in the global configuration. The post processing script must be executable and accept the path of the exported CSV file as the first argument.

Example post processing scripts are provided in the scripts sub directory located with the ISP Billing Script distribution. The following files are provided as examples:

- post_process.pl – General example post processing script to start with, written in perl.
- post_process_tab.pl – Post processor that converts CSV to TAB delimited file, written in perl.
- post_process_xml.pl – Post processor creates xml export file, written in perl.
- post_process_platypus_example.pl – Example post processor creates an importable CSV file for Playtpus Billing System, written in perl. This script needs to be altered to fit your billing profiles in Playtpus.
- post_process_sql_example.pl – Example post processor that creates SQL insert statements, written in perl.

Global File Export fields:

XML Field	Description
<global>	Globals opening tag
<defaults>	Defaults opening tag
<file process=”yes”></file>	Name of the file to write CSV data. Parameter “process” can be set to “no” to disable a particular customer or globally running the post processing script. Default value of “yes” is assigned to the “process” parameter if it is not defined.
</defaults>	Defaults closing tag
<export_file_processor> </export_file_processor>	Post processing script that is executed after the export file has been written. The first parameter that is provided to the script is the path to the exported CSV data file.
</globals>	Globals closing tag

Example Globals File Export XML Parameters:

```
<global>
  <export_file_processor></export_file_processor>
  <defaults>
    <file process="yes">export.csv</file>
    <email>NULL</email>
  </defaults>
</global>
```

Note: These are optional parameters and should be added to the appropriate sections of your configuration.

Threshold Tracking

Sub Section ix

With version 1.0.10 of the ISP Billing Script there comes a very complex feature that allows you to inform you customers at what time an overage (exceed committed rate) occurred. While some may think that this is an easy task, and it is for Bandwidth Summation, it however is not for Nth Percentile. The following options when enabled will allow for overage notifications to be sent to customers as well as reporting to the customer at what time an overage occurred.

Enabling threshold tracking will use more memory and processor power. It is strongly suggested that you follow best practices by running the ISP Billing Script every day. This way, the CPU usage to calculate time frame threshold data will occur over everyday of the billing period.

Global Threshold Tracking fields:

XML Field	Description
<global>	Globals opening tag
<threshold>	Threshold opening tag
<enabled></enabled>	Enable threshold tracking for notification on billing reports of when overage occurred. Default “no”, set to “yes” to activate.
<notification></notification>	Enable threshold notifications emails when customers exceed their committed rate. Default “no”, set to “yes” to enable.
<subject></subject>	Subject line to set for notification emails.
<title></title>	Title to set the title and name of the from email for notification emails.
</threshold>	Threshold closing tag
</globals>	Globals closing tag

NOTE: Because of the nature of Nth Percentile calculations, a customer could be notified multiple times about potentially going over their committed rate, but in the end they may not go over their committed rate. Because of that the Nth Percentile notifications are limited to sending out one if an overage is detected, making note and not sending out another notification until the Nth percentile value has fallen below the threshold and risen above it again. A maximum of 4 notifications per billing period will be sent, equally spaced over the billing period.

NOTE: Notifications are limited to once a day, and are notifications of overages occurring on the prior day.

Example Globals Threshold Tracking XML Parameters:

```
<global>
  <threshold>
    <enabled>yes</enabled>
    <notification>yes</notification>
    <subject>Email Notification Subject</subject>
    <title>Email Address Title</title>
  </threshold>
</global>
```

Note: These are optional parameters and should be added to the appropriate sections of your configuration.

Using the build Command

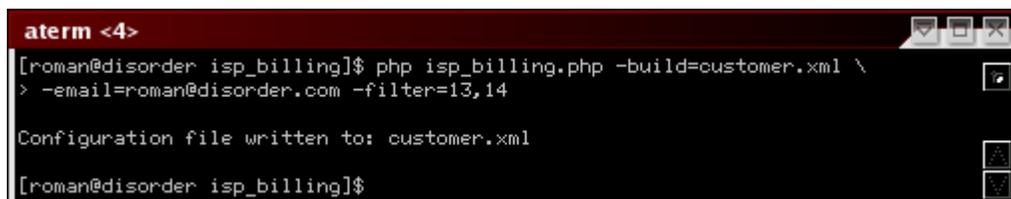
Sub Section x

The build command line option allows you to build an XML configuration file for your customers. Some modification of the XML configuration file will be needed after it is created. The rates, billing intervals and the customer descriptions should be reviewed and edited.

The following command line options are available to the build command:

Option	Description
build	Required to build a configuration file. If file name is defined, then the configuration will be written to the defined file.
filter	Used to limit the built configuration file to a list of graph id's. Graph id's are comma separated string with no spaces.
email	If defined, sets the default email address in the built configuration file.
email_no_html	If defined, sets the default email to not send html billing reports
email_no_csv	If defined, set the default email to no send csv attachments on billing reports

Example 11 – Build command line



```
aterm <4>
[roman@disorder isp_billing]$ php isp_billing.php -build=customer.xml \
> -email=roman@disorder.com -filter=13,14
Configuration file written to: customer.xml
[roman@disorder isp_billing]$
```

Example Configuration

Sub Section xi

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<cacti_biller>
  <global>
    <defaults>
      <email>billing@isp.com</email>
      <rate unit="Mb">0.05</rate>
      <billing_timeframe>
        <type>monthly</type>
        <day>last</day>
        <every>1</every>
        <start_date>2006-10-01</start_date>
      </billing_timeframe>
    </defaults>
  </global>
  <customers>
    <customer>
      <description>Firewall - Traffic - 192.168.1.1 (PIX Firewall in) – 1</description>
      <email html="no">billing@isp.com</email>
      <email csv="no">customer@isp.com</email>
      <rate unit="Mb">1.00</rate>
      <billing_timeframe>
        <type>bi-monthly</type>
        <start_date>2006-09-21</start_date>
      </billing_timeframe>
      <graphs>
        <graph>
          <id>14</id><!-- Graph Title: "Firewall - Traffic - 192.168.1.1 (PIX Firewall in)" -->
          <graph_item_id>509</graph_item_id><!-- Comment: "Total Out: |sum:auto:current:2:auto|"-->
          <graph_item_id>510</graph_item_id><!-- Comment: "Total In: |sum:auto:current:2:auto|"-->
          <rate unit="Mb" type="committed">0.50</rate>
          <rate unit="Mb" type="overage" threshold="1000000000">2.00</rate>
        </graph>
        <graph>
          <id>16</id><!-- Graph Title: "Firewall - Traffic - 192.168.1.1 (PIX Firewall in)" -->
        </graph>
      </graphs>
    </customer>
  </customers>
</cacti_biller>
```

Track File

Sub Section xii

Question: What is the track file?

Answer: The track file is a xml file that contains information when a customers billing report was last successfully generated.

Question: Do I need to create a track file?

Answer: No, the track file will be created the first time that something needs to be written to it.

Question: Why does the ISP Billing Script require the “-track=” option for the “-info” command?

Answer: Because if the file exists, it will show the information stored in the track file in the output of the “-info” command.

Question: What is the format of the track file?

Answer: While not really that important to anyone using the ISP Billing Script, the format is very simple XML records that contain the customer description and timestamp of the last billing cycle end time. If threshold tracking is enabled, this file will also contain threshold tracking cache information.

Testing

Section 7

So, you have your configuration created for all your customer and you are ready to test it. This section will explain how testing can be performed and what considerations should be known when testing.

Example testing command line:

```
php isp_billing.php -d -config=config/customers.xml -track=config/track.xml -track_no_write  
-start_date="2007-02-01 00:00:00" -current_time="2007-03-01 00:00:01"
```

The above example would be used to test a monthly billing customer. The start date would be overridden with the first day of February, "2007-02-01 00:00:00" and the current time used by the billing script would be overridden with first day of March, "2007-03-01 00:00:00".

Note: Important to understand that the current time override is an internal variable in the billing script. Do not define the last day of February, because in normal operation, the billing script needs to wait till the time frame that it is billing for has past. This means that you must define a current time one day from the last day of the billing interval you are testing.

The "-d" command line parameter is important to have while testing. This will output debug about what the billing script is doing. Depending on your configuration, this could be a lot of information.

CSV File Formats

Section 8

Email CSV File

Sub Section i

The following table outlines the fields in the CSV file in the order which they appear.

Field	Description
Customer Description	Description of customer
Billing Period Start	Start of billing period
Billing Period End	End of billing period
Graph Title	Title of Graph
Type	Type of Cacti graph variable
Interval	Billing interval
Every	Billing interval repeat factor
Rate Type	Rate type
Rate Amount	Per unit rate amount
Rate Unit	Rate unit
Bits	Number of bits used in billing calculation
Total	Calculated total
Overage Occurred	If threshold tracking is enabled, this will have when the overage occurred

Export CSV File

Sub Section ii

The following table outlines the fields in the CSV file in the order which they appear.

Field	Description
CustomerDescription	Description of customer
CustomerExternalId	External id defined for customer
BillingPeriodStart	Start of billing period
BillingPeriodEnd	End of billing period
GraphId	Cacti graph id
GraphTitle	Title of graph
GraphExternalId	External id defined for graph
GraphItemId	Cacti graph item id
Type	Type of Cacti graph variable
Interval	Billing interval
Every	Billing interval repeat factor
RateType	Rate type
RateAmount	Per unit rate amount
RateUnit	Rate unit
Bits	Number of bits used in billing calculation
Total	Calculated total
OverageOccurred	If threshold tracking is enabled, this will have when the overage occurred

Command Line Reference

Section 9

```
Cacti ISP Billing Script, Copyright 2006-2008 - The Cacti Group
Version: 1.0.9
usage: -config=[file] -track=[file] [-check] [-build=[file] [-filter=[id]]] [-process] [-list]
      [-email=[email]] [-email_no_html] [-html_no_csv] [-start_date=[date]]
      [-current_time=[date]] [-tech] [-d] [--debug] [-h] [--help] [-v] [--version]

-config=[file]    - Billing configuration file
-track=[file]     - Date tracking file
-track_no_write  - Do not update the date tracking file
-track_clear_cache - Clear threshold tracking cache from track file
-check           - Check billing configuration file
-build=[file]    - Build example configuration file from system, supplying filename is
                  optional, default example.xml
-filter=[id]     - Only used by the build command to limit configuration build to the supplied
                  graph ids, comma delimited
-info           - Display information on the billing configuration file
-list           - Display list of graphs and titles that are billable
-email=[email]   - Override configuration email addresses, all customer reports will be
                  emailed to the supplied email
-email_no_html  - Only used when email override enabled, globally set no html emails
-email_no_csv   - Only used when email override enabled, globally set no csv attachments
-start_date=[date] - Used to override track date and start date for testing and reruns
-current_time=[date] - Used to override current time for testing and reruns
-tech           - Writes out technical support file
-d --debug      - Display verbose output during execution
-v --version    - Display this help message
-h --help      - Display this help message
```

There are many command line options, but only two are required for normal operations “config” and “track”. Outline below are detailed explanations for each displayed command line option.

Option	Description
config=	Path to the customer configuration XML file
track=	Path to the track XML file. This file is used to track the last time a customer was processed. The description defined for the customer is what is used for tracking.
track_no_write	Disables updating the tracking file.
track_clear_cache	Clears threshold tracking cache data from track file and exits.
check	Checks the configuration file for issues.
build	Builds a configuration file from the graphs configured in Cacti that are considered billable. Optional filename can be supplied. Use the filter option to limit what graphs are created.
filter	List of graph ids to build a configuration. This is only used by the build command line option.
info	Displays information about the configured customers.

Option	Description
list	Displays a list of graphs configured in Cacti that are considered billable.
email	Overriding email address for processing. When processing customers and this is defined, all configured customers will be emailed to the defined email. This parameter is also used when building configuration with the build command line option.
email_no_html	Only used when email override is enabled, turns off HTML formatted email reports, making them text only.
email_no_csv	Only used when email override is enabled, turns off the attachment of the CSV file to email reports.
start_date=	Override the track file and configuration start dates for customers. Used for testing configurations.
current_time=	Override the current time used by the billing script. Used for testing configurations.
tech	Writes out technical support file, used by technical support to get system information
d or debug	Debug mode, outputs verbose information about processing
v or version	Displays help and version
h or help	Displays help and version