Automatic periodic increase credit LIMIT

Document created on [2014-10-23](http://switzernet.com/3/public/141022-periodic-increase-credit-limit/index1.docx)
Nicolas Bondier

[[pdf](index.pdf)][[doc](index.docx)][[htm](index.htm)]

Contents

[Introduction 3](#_Toc401834668)

[Script documentation 4](#_Toc401834669)

[Liens 13](#_Toc401834670)

# Introduction

This document describes the script used to periodically increase the credit limit of the customers. It is on slave server in folder /home/porta-admin/141022-periodic-increase-credit-limit.

# Script documentation

|  |  |
| --- | --- |
| CODE | COMMENT |
| #!/usr/bin/env perl**use** strict;**use** warnings;**use** DBI;**use** SOAP::Lite# +trace=>'debug';**use** File::Spec::Functions qw(rel2abs);**use** File::Basename;**use** MIME::Lite;**use** Config::IniFiles;**use** POSIX qw(strftime);**use** Data::Dumper;**use** File::stat;**use** Archive::Tar; | Libraries the script uses. |
| **my** $soap\_login = '';**my** $soap\_password = '';**my** $cfg = Config::IniFiles->**new**( -file => "/home/porta-admin/141022-soap-credentials/soap.ini" );**if** ( $cfg->val( 'admin', 'login' ) && $cfg->val( 'admin', 'password' ) ){ $soap\_login = $cfg->val( 'admin', 'login' ); $soap\_password = $cfg->val( 'admin', 'password' );} **else** { exit;} | Getting SOAP credentials from configuration file |
| **my** $dirname = dirname(rel2abs($0));chdir($dirname);**my** $log\_path = $dirname . '/log/';**my** $logfile = "periodic-increase-credit-limit.log";**my** $logfile\_archive = strftime "%Y-%m-%d\_$logfile", localtime;**my** $logfile\_zip = "$logfile\_archive.tar.gz";$logfile = $log\_path . "/" . $logfile;$logfile\_archive = $log\_path . "/" . $logfile\_archive;$logfile\_zip = $log\_path . "/" . $logfile\_zip;**if** ( -e $logfile ){ **my** $last\_modification\_day = sprintf ("%d", stat($logfile)->mtime / (60\*60\*24) ); **my** $today = sprintf( "%d", time / (60\*60\*24) ); **if** ( $last\_modification\_day != $today && stat($logfile)->size > 0 ){ **print** "\nCompressing old file ..."; rename $logfile, $logfile\_zip; **my** $tar = Archive::Tar->**new**; $tar->write( $logfile\_zip, COMPRESS\_GZIP ); **print** "OK"; }}**print** "\n";open (LOGFILE, '>>' .$logfile);  | Creation of a daily rotating log.The old logs are compressed and stored by date. |
| **my** $host = 'xxxxxxxxxxxxxxxxxx';**my** $port = '3306';**my** $database = 'porta-billing';**my** $login = 'xxxx';**my** $password = '';**my** $dbh = DBI->connect( "DBI:mysql:database=$database;host=$host;port=$port", $login, $password ) or die "Connection impossible à la base de données $database !\n $! \n $@\n$DBI::errstr"; | Connection to the master database. |
| # SOAP CONNECTION TO NEW BILLINGbinmode(STDOUT, ':utf8');**my** $proxy\_host = 'https://slave.switzernet.com'; # Porta-Billing Admin Server**my** $proxy\_port = '8444';**my** $uri\_base = 'http://portaone.com/Porta/SOAP';**my** $proxy = "$proxy\_host:$proxy\_port/soap/";**my** %uris = ( 'Session' => "$uri\_base/Session", 'Account' => "$uri\_base/Account", 'Customer' => "$uri\_base/Customer",);**sub** fault\_handler { **my** ($soap, $res) = @\_; die "SOAP Fault: $!, " . (ref $res ? $res->faultstring : $soap->transport->status);}**my** $session\_service = SOAP::Lite ->uri($uris{'Session'}) ->proxy($proxy) ->on\_fault(\&fault\_handler);**my** $customer\_service = SOAP::Lite ->uri($uris{'Customer'}) ->proxy($proxy) ->on\_fault(\&fault\_handler);**my** $account\_service = SOAP::Lite ->uri($uris{'Account'}) ->proxy($proxy) ->on\_fault(\&fault\_handler);# required to support dateTime type$session\_service->serializer() ->xmlschema('http://www.w3.org/2001/XMLSchema');$customer\_service->serializer() ->xmlschema('http://www.w3.org/2001/XMLSchema');$account\_service->serializer() ->xmlschema('http://www.w3.org/2001/XMLSchema');**my** $LoginResponse = $session\_service->login($soap\_login, $soap\_password);**my** $session\_id = $LoginResponse->result();**print** "Logged in with session $session\_id\n";**my** $header = SOAP::Header->name('auth\_info')->value({ session\_id => $session\_id }); | Connection to Porta-Billing SOAP. |
| **my** $max\_credit\_limit = 750;**my** $req = '';$req = 'SELECT c.i\_customer, c.name, c.firstname, c.lastname, c.email, c.credit\_limit, IF(2\*c.credit\_limit<2\*i.previous\_balance, if( 2\*c.credit\_limit>=' . $max\_credit\_limit . ',' . $max\_credit\_limit . ', truncate(2\*c.credit\_limit,-1 )), if(2\*i.previous\_balance >=' . $max\_credit\_limit . ',' . $max\_credit\_limit . ', truncate(2\*i.previous\_balance,-1) ) ) as "newlimit", c.iso\_4217, cn.notepadFROM Customers cINNER JOIN Invoices iON  c.i\_customer=i.i\_customerINNER JOIN Customer\_Notepad cnON cn.i\_customer=c.i\_customerWHERE c.i\_rep=3AND c.credit\_limit>50AND i.previous\_balance+i.payments<=0AND 100\*(c.balance/c.credit\_limit)>=70AND c.credit\_limit<2\*i.previous\_balanceAND i.issue\_date>=(date\_sub(now(), interval 1 month))HAVING newlimit>c.credit\_limit'; | The request for finding all customers who need to increase their credit limit. |
| **my** $customers = {};$customers = $dbh->selectall\_hashref($req, 'i\_customer');**foreach** **my** $i\_customer (keys %$customers) { **my** $line = ''; **foreach** **my** $keys (keys %{ $customers->{$i\_customer} } ) { $line = $line . $customers->{$i\_customer}->{$keys}.","; } chop($line); **print** LOGFILE '[', strftime("%Y-%m-%d %H:%M:%S", localtime) , '] ', $line , "\n";} | Printing all lines to the log file in order to backup data before we modify it. |
| **my** $body\_html = <<EMAIL\_BODY\_HTML;<html> <head> <style> </style> </head> <body><p>-- English version below</p><p>Chère/Cher [[firstname]] [[lastname]],</p><p>En tenant compte de votre consommation téléphonique et de vos paiements du mois passé, nous avons augmenté la limite de crédit de votre compte.<br>Elle était à [[oldlimit]] [[currency\_code]], et passe maintenant à [[newlimit]] [[currency\_code]].</p><p>Meilleures salutations,<br>L’équipe Switzernet</p><p>----</p><p>Dear Customer,</p><p>After considering your phone line usage and your payments of the last month, we increased the credit limit of your account.<br>It was set to [[oldlimit]] [[currency\_code]], and is now [[newlimit]] [[currency\_code]].</p><p>Best regards,<br>The Switzernet team</p> </body></html>EMAIL\_BODY\_HTML | The email body. |
| **foreach** **my** $i\_customer (keys %$customers) { **my** $customer\_body = $body\_html; **my** $newlimit = $customers->{$i\_customer}->{newlimit}; **my** $oldlimit = $customers->{$i\_customer}->{credit\_limit}; **my** $currency\_code = $customers->{$i\_customer}->{iso\_4217}; **my** $firstname = $customers->{$i\_customer}->{firstname}; **my** $lastname = $customers->{$i\_customer}->{lastname}; | For each record found with the MySQL request we get the necessary values for the mail and notepad. |
|  $newlimit = sprintf "%01.2f", $newlimit; $oldlimit = sprintf "%01.2f", $oldlimit; **my** $date = strftime "%y%m%d", localtime; **my** $add\_to\_notepad = '<limit on='.$date.' from='.$oldlimit.' to='.$newlimit.' why=history />'; | The credit limits are formatted with to decimal.The data must match the notepad format.Creation of the new notepad line. |
|  $customer\_body =~ s/\[\[newlimit\]\]/$newlimit/g; $customer\_body =~ s/\[\[oldlimit\]\]/$oldlimit/g; $customer\_body =~ s/\[\[firstname\]\]/$firstname/g; $customer\_body =~ s/\[\[lastname\]\]/$lastname/g; $customer\_body =~ s/\[\[currency\_code\]\]/$currency\_code/g; | Replacing the keywords in the customer body with the good values. |
|  **my** $CustomerInfo = { 'i\_customer' => $i\_customer, 'credit\_limit' => $customers->{$i\_customer}->{newlimit} }; **my** $UpdateCustomerRequest = { 'customer\_info' => $CustomerInfo }; **my** $AddUpdateCustomerResponse = $customer\_service->update\_customer( $header , $UpdateCustomerRequest )->result; | Updating the customer limit with new value. |
|  **if** ( defined( $AddUpdateCustomerResponse->{i\_customer} ) && $AddUpdateCustomerResponse->{i\_customer} =~ /^[0-9]+$/ && $AddUpdateCustomerResponse->{i\_customer} =~ /^[0-9]+$/ > 0 ){ **my** $sth = $dbh->prepare('UPDATE Customer\_Notepad SET notepad = CONCAT( ? , "\n" , notepad ) WHERE i\_customer = ?') or die "Couldn't prepare statement: " . $dbh->errstr; $sth->execute($add\_to\_notepad, $i\_customer) or die "Couldn't execute statement: " . $sth->errstr; } | If the SOAP request return the i\_customer, then we can update the database. |
|  **print** "\nsending to ".$customers->{$i\_customer}->{email}; **my** $msg = MIME::Lite->**new**( From => 'user@domain.com', To => $customers->{$i\_customer}->{email}, Cc => 'user@domain.com', Subject => 'Augmentation de votre limite de crédit', Type => 'multipart/mixed' ); $msg->attach( Type => 'text/html; charset=UTF-8', Data => $customer\_body ); $msg->send;} | Sending the email info to internal and customer address. |
| $dbh->disconnect();close (LOGFILE);**print** "\nEnd of the script\n"; | Closign database and log file. |

# Liens

Ce document :

[http://switzernet.com/3/public/141015-automatic-web-subscriptions/](http://switzernet.com/3/public/folder/)

\* \* \*



Copyright © 2014 by Switzernet