JPost Local (CGI)

Technical Document

v.2, October, 2016

Disclaimer

CCBill shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of this documentation. In no event shall CCBill be liable for any consequential, indirect, special or incidental damages, even if provider has been advised by user of the possibility of such potential loss or damage. User agrees to hold provider harmless from and against any and all claims, losses, liabilities and expenses.

You may not distribute, reproduce, transmit or transfer, in whole or in part, the documentation contained herein without express prior written permission from the provider. CCBill reserves the right to revise and to make changes to this manual without incurring any obligation to notify any person of such changes or revisions.

Contents

Introduction

This document is provided as a technical resource to CCBill merchants.

JPost Local, a Common Gateway Interface (CGI), is an Application Program Interface (API) to the CCBill User Management Network. The primary purpose of the JPost Local is to receive information and instructions from the CCBill Network and execute functions in regard to the User Management System.

The information contained in this document outlines the JPost Local functionality and its interaction with the CCBill User Management System.

1. CCBill User Management System

The CCBill, LLC User Management System is a portion of the Signup Forms system and is used by CCBill merchants to manage various merchant side authentication systems. The user management system is an event driven system. When an event is initiated the user management system will send data and instructions to the merchant side API (Application Program Interface). Upon receiving instructions from the CCBill network the merchant side API will execute different functions on the data received.

1.1 Server Side Events

1.1.1 Add/Update Username Events

There are several events that will issue an ADD or an UPDATE command to the merchant side API including the signup process and manual add. During the signup process the user management system will issue two different commands to the merchant side API: **ADD** and **UPDATE**.

The ADD command is issued prior to billing the consumer and instructs the merchant side API to add a username and password to the merchant's authentication system. During the reservation portion of the process the ADD command is issued to the API with the consumer's valid username and an encrypted password unknown to the consumer. This process is completed to ensure that the username and password chosen by the consumer are valid and can be used within the merchant's authentication system.

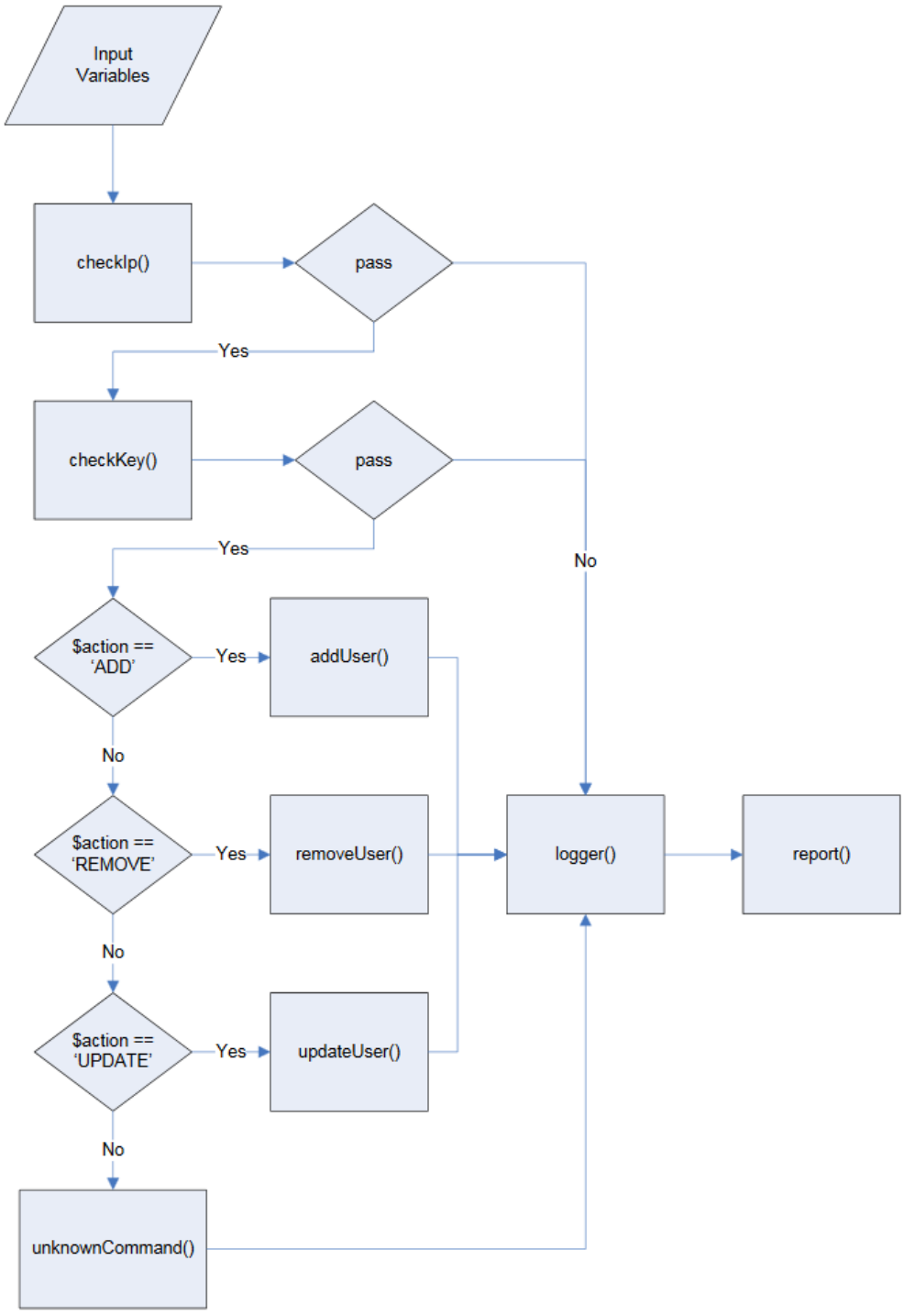
Upon receiving a positive response back from the merchant-side API indicating that the reservation was completed successfully the CCBill Forms system will attempt to bill the consumer. If the consumer is properly billed, the user management system will issue **the UPDATE command** to the merchant side API and instruct the merchant side API to update the record for the valid username with the valid password. The consumer has now been given access to content protected by the merchant's authentication system.

1.1.2 Remove Username Events

Several events can cause the user management system to attempt to remove a username and password from the merchant's authentication system. Such events include normal expiration and manual remove.

2. The JPost Local API

2.1 System Flowchart



2.2 Dependencies

2.2.1 Digest::MD5 'md5_hex'

The Digest::MD5 module is a common CPan module that usually comes installed as a native module with most distributions of Perl.

2.3 Constants

2.3.1 Response Codes:

- **$VERSION**
  This is the version of the API produced by CCBill that is executing on the merchant's server. The most recent version of the API is 1.1.0.
- **$FAILURE - 000**
  The $FAILURE response code is produced after an action has been submitted to the API and the API has completely failed to complete the instruction. The value of this constant is 000.
- **$DUPLICATE_USER - 001**
  A response code issued back to the CCBill network after receiving an 'ADD' command indicating that the username received to be added was already found within the merchant's authentication system and is being used by a different user. Upon receiving the response the CCBill Forms System will prompt the consumer to choose a different username. The value is 001.
- **$USER_NO_EXIST – 010**
  A response code issued back to the CCBill network after receiving a 'REMOVE' command indicating that the username received does not exist within the merchant's authentication system. The value is 010.
- **$BAD_PERMISSIONS – 011**
  A response code issued back to the CCBill network after receiving any command indicating that the action could not be completed because the API was unable to write to one or more of the required system files due to a permissions error. The response will inform the CCBill Merchant Support Department of a merchant side system failure and they will be prompted to resolve the problem. The value is 011.
- **BAD_CHECKSUM – 100**
  A response code issued back to the CCBill network indicating that the API posted to does not host merchant based information hosted on the CCBill network. This response is a security error and will prompt the CCBill Merchant Support Department to resolve the problem. The value is 100.
- **FATAL_ERROR – 101**
  A response code issued back to the CCBill network indicating that information received by the API was insufficient or invalid; e.g. NULL values for input (username, password, reservationId). An interpreted response will cause the CCBill system to work in different ways depending on the command issued. The value is 101.
- **BAD_IP – 110**
  A response produced by the merchant API indicating that an action received was not received by the CCBill network. This response will cause the system to log an instance of a possible security breach attempt by the offending IP Address. The value is 110.

2.3.2 System Constants:

- **$ACTION_LOG_FILE**
  The path from server root "/" or web root to the action log file. The action log file is a flat text file to which all actions completed by the API are recorded. The API must have read/write privileges to the action log file.
- **$ERROR_LOG_FILE**
  The path from server root "/" or web root to the error log file. The error log file is a flat text file to which all errors encountered by the API during completing an action are recorded. The API must have read/write privileges to the error log file.

- **$PASSWORD_FILE**
  The path from server root "/" or web root to the authentication system. The $PASSWORD_FILE constant is used for those APIs that support writing authentication information to a flat text file. The API must have read/write privileges to the password file.

2.4 Input Variables

2.4.1 The Input Hash (%in)

The input has named *%in* is a data hash to which all of the input name value pairs are stored. Naming conventions are key = name and value = value for the %in hash. **See sub-routing readParse()*.

2.4.2 $key

The $key value is an MD5 value that is passed to the API by the CCBill network to ensure that the command is being issued to the correct API. The MD5 hash consists of an md5_hex encryption of the $username, $action, and $PRIVATE_KEY constant concatenated into one string. The result should match the $key value.

2.4.3 $action

The action variable is the command value issued by the CCBill network to the API. Possible values include ADD, REMOVE, UPDATE, and VERSION. The API will act according to the value of the $action input variable.

2.4.4 $username

The username variable is the value of the username to perform the received action on.

2.4.5 $password

The password variable is that which matches the username received by the API.

2.4.6 $reservationId

The reservationId variable is that which matches the username received by the API and is used for logging purposes only.

2.5 Sub-Routines

2.5.1 addUser($$$)

- **Synopsis**

The addUser sub-routine receives variables and attempts to add a username and password to the merchant's authentication system. Upon a successful add the API will log the event and respond with the appropriate response code. If the action could not be performed the API will log the failure and report a failure.

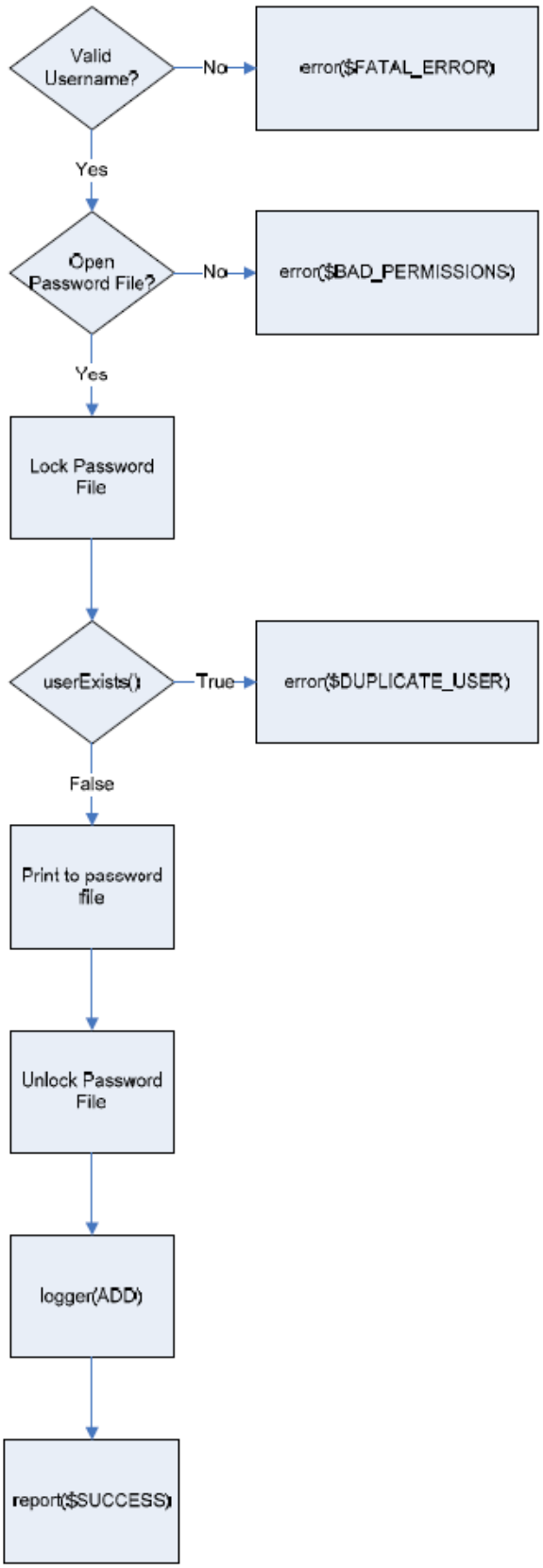
- **Arguments**

  • $username

  • $password

  • $reservationId

- **addUser($$$) Process Flowchart**



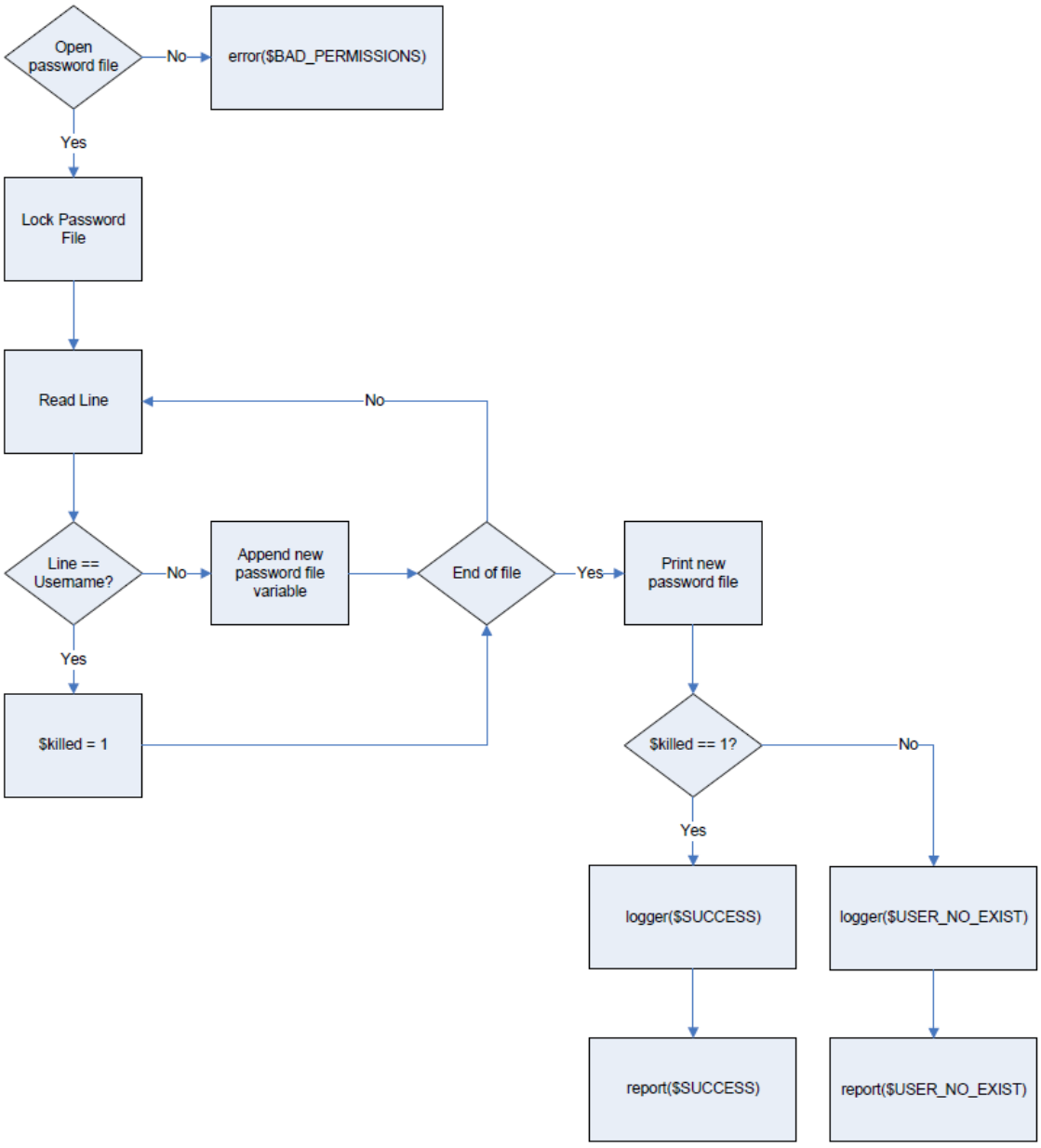
2.5.2 removeUser($$)

- **Synopsis**

The removeUser sub-routine receives data from the CCBill network and attempts to remove the provided username and password. Upon a either completing or failing to complete the action the subroutine will attempt to log the action and respond with the appropriate response.

- **Argument**
  - $username
  - $reservationId

- **removeUser($$) Process Flowchart**



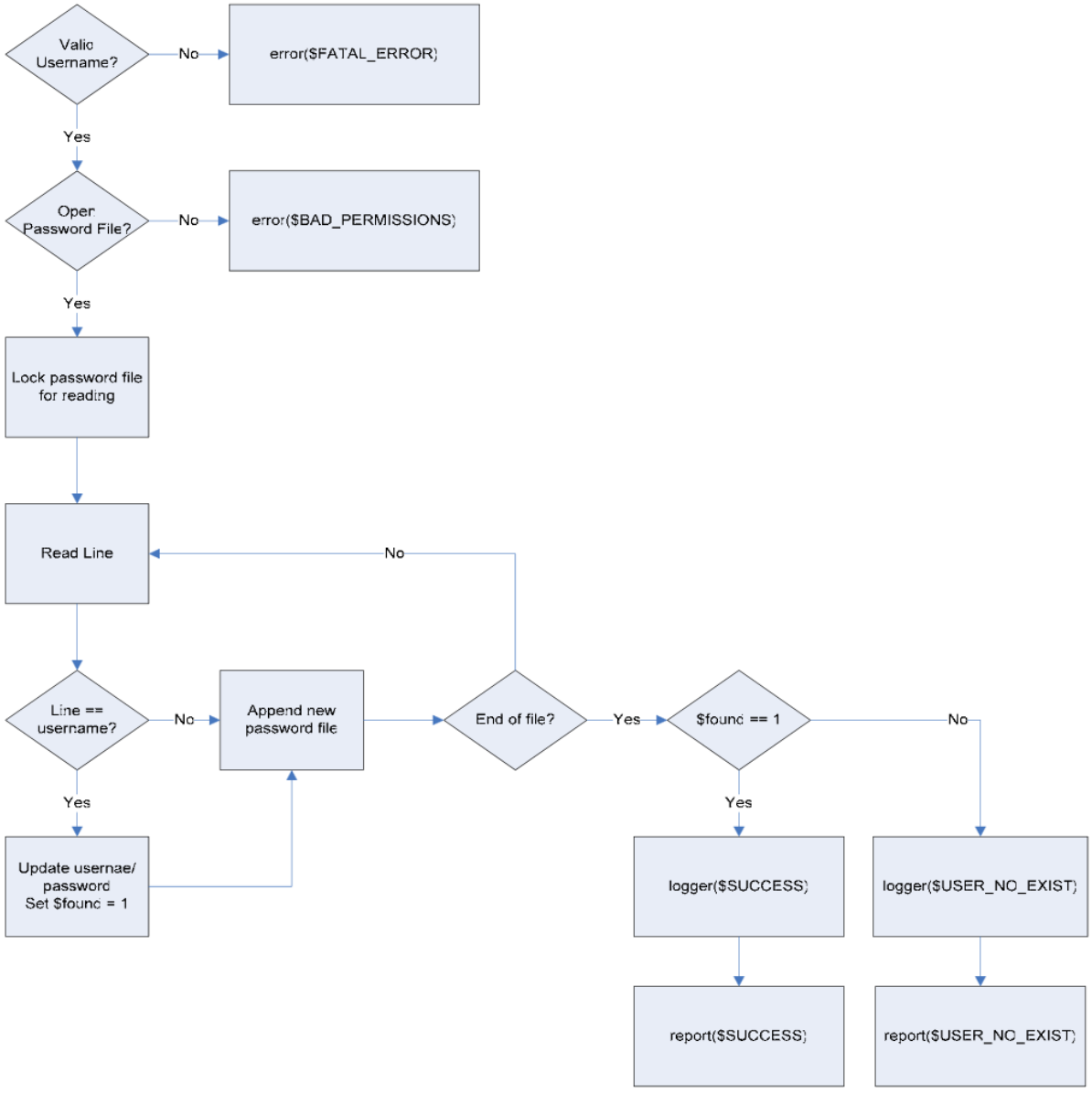
2.5.3 updateUser($$$)

- **Synopsis**

The updateUser sub-routine receives data from the CCBill, LLC network, searches through the merchant's authentication system for a matching username, and updates the username's corresponding password with the new value passed into the API. This function is called for such actions as update reservation, changing a username and password, etc.

- **Arguments**
  - $username
  - $password
  - $reservationId

- **updateUser($$$) Process Flowchart**



2.5.4 userExists($$)

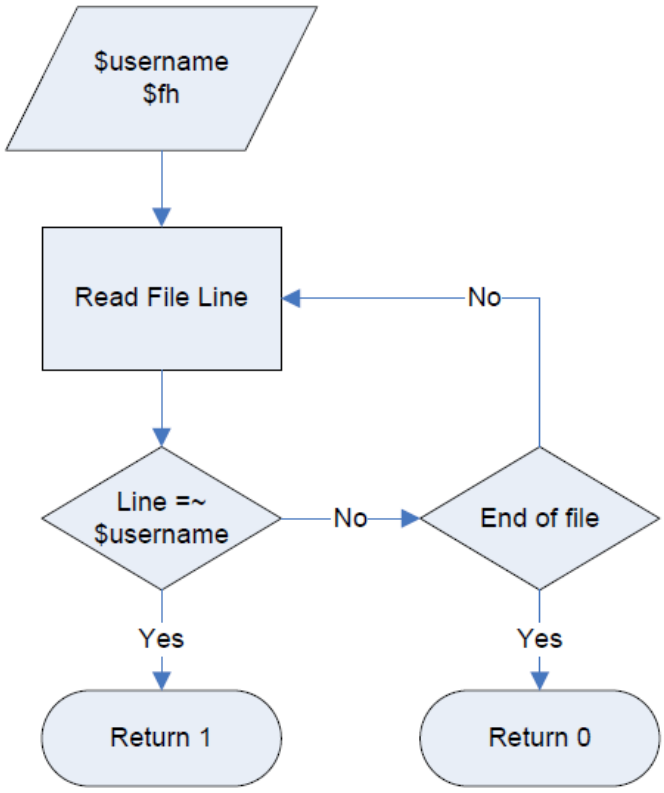
- **Synopsis**

  The user exists sub-routine receives a username value and a file handle. The sub-routine loops through the file handle reading each line of text and does a string compare against the username value and the line value. If the username and the line value match the sub-routine returns a 1 otherwise the sub-routine will return a 0. The userExists sub-routine is a Boolean function.

- **Arguments**

  - $username
  - $fh

- **userExists($$) Process Flowchart**

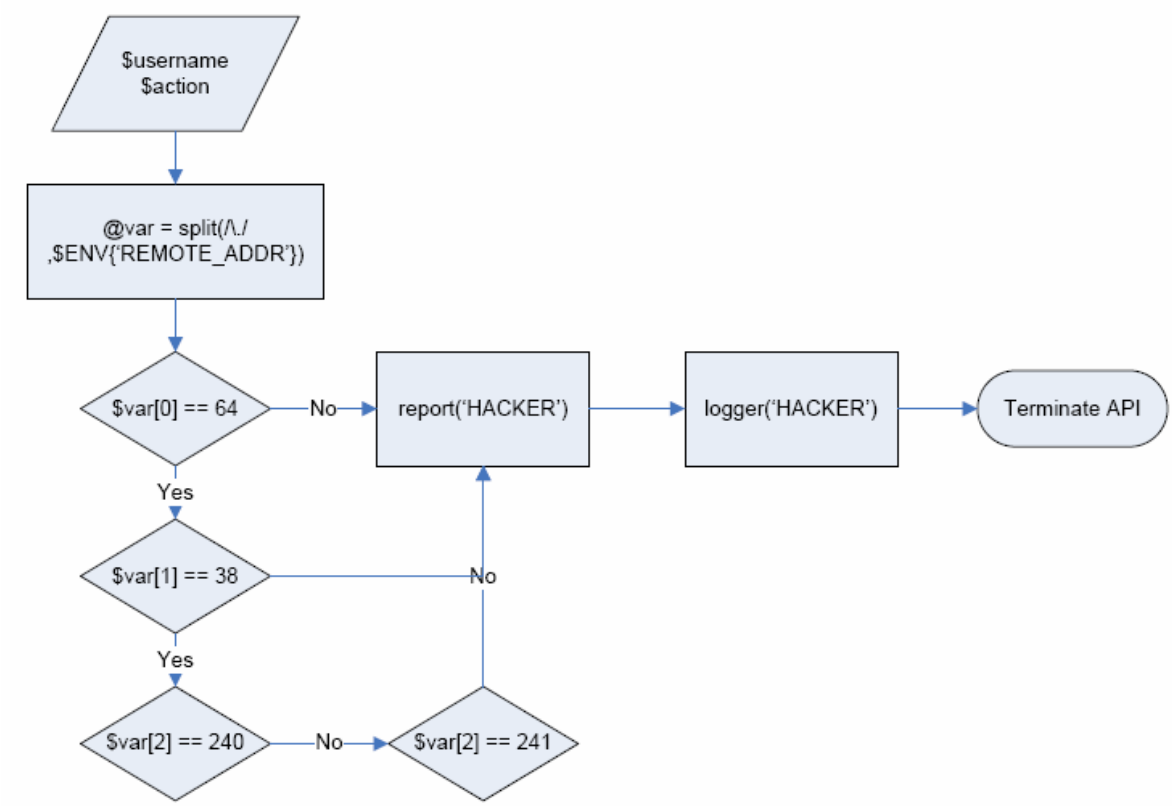


2.5.5 checkIP($$)

- **Synopsis**

  The checkIP sub-routine receives the $action and $username input parameters as arguments. The sub-routine reads the REMOTE_ADDR environment variable and checks it against the CCBill network IP address which is hard coded into the sub-routine. If the REMOTE_ADDR condition fails the sub-routine runs the report() and logger() sub-routines and terminates the API. If the REMOTE_ADDR condition passes the API continues top down processing.

- **checkIP($$) Process Flowchart**



2.5.6 checkKey($$;$)

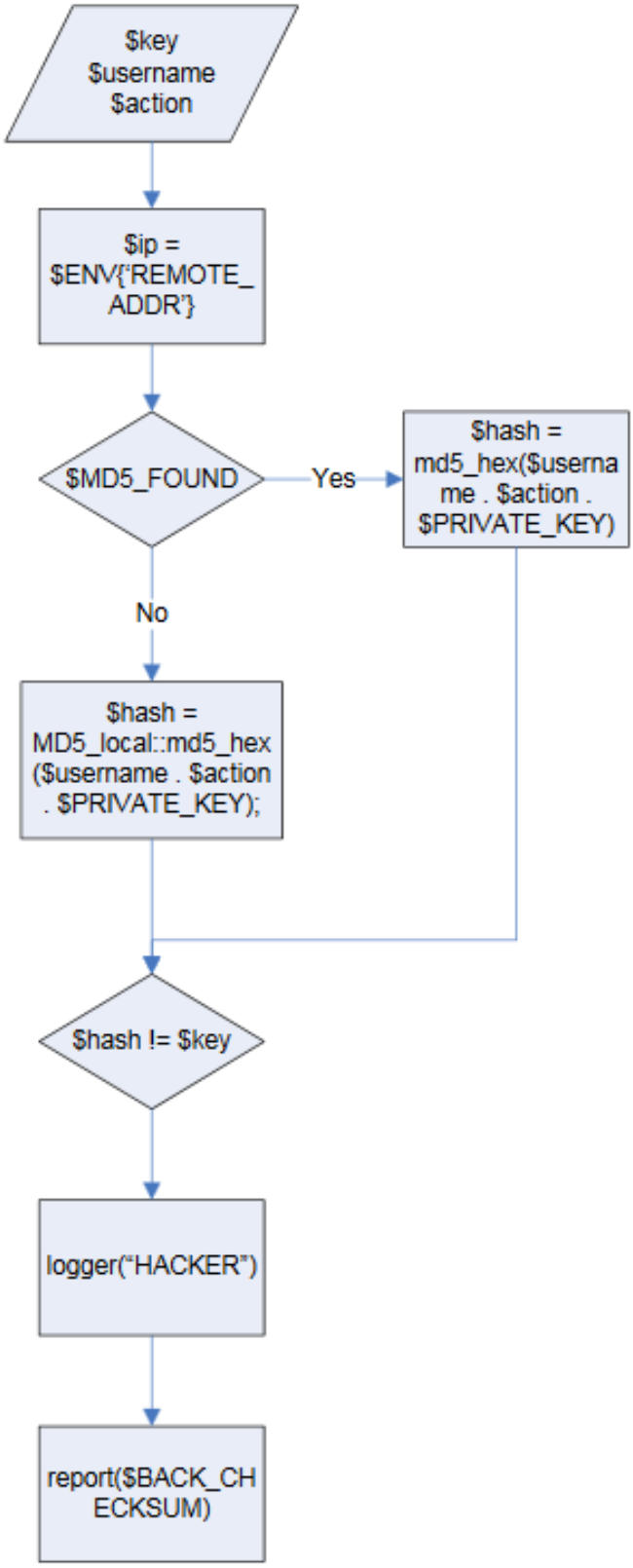
- **Synopsis**

  The checkKey sub-routine is a security measure function which compares an MD5 ::md5_hex encryption salted with the $username, $action, and $PRIVATE_KEY values. The compare is done against the $key input parameter that is received as an argument. If the string compare does not match the sub-routine calls report() and logger() and terminates the API. If the string compare results to true the API continues processing.

- **Arguments**

  - $key
  - $username
  - $action

- **checkKey($$;$) Process Flowchart**



2.5.7 error($$;$)

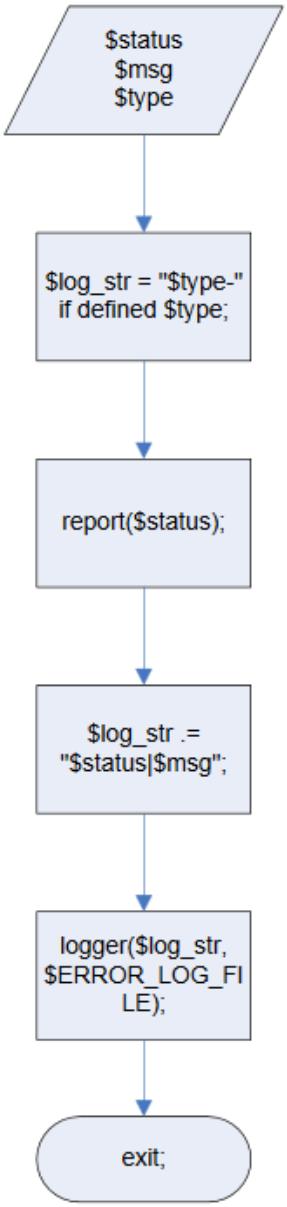
- **Synopsis**

  The error sub-routine calls both the logger and report functions passing the arguments received as parameters to both functions and then terminates the API from processing.

- **Arguments**

  - $status
  - $msg
  - $type

- **error($$;$) Process Flowchart**



Online Definitions

Below are a few websites that can be used to find definitions, terms and acronyms you want to be more familiar with. By clicking on the links you will be redirected to the webpage in question.

The Acronym Finder is a general-purpose reference of common acronyms. Access up 126,600 acronyms/ abbreviations and their meanings. It provides a searchable database containing common acronyms and abbreviations about computers, technology and military subjects.

NetLingo, the Internet Language Dictionary, received a "Best of Web" award and has a searchable format. It contains an Internet language dictionary and you can search either alphabetically or by keyword. It contains a comprehension list of terms and definitions and provides links to pertinent information.

Webopedia, Online Computer Dictionary for Internet Terms and Technical Support. This is an online dictionary for computer and Internet related information. It has the capability to search by keyword or category. It contains links to other related Internet information. Some of the features of this site are as follows: computer dictionary, Internet encyclopedia, computer terms, Internet search engine, and computer education.