



# PortaSwitch

## External System Interfaces Guide

75

MAINTENANCE  
RELEASE

## **Copyright notice & disclaimers**

Copyright © 2000–2019 PortaOne, Inc. All rights reserved

**PortaSwitch® External System Interfaces, February 2019**  
**Maintenance Release 75**  
**V1.75.03**

Please address your comments and suggestions to: Sales Department,  
PortaOne, Inc. Suite #408, 2963 Glen Drive, Coquitlam BC V3B 2P7  
Canada.

Changes may be made periodically to the information in this publication. The changes will be incorporated in new editions of the guide. The software described in this document is furnished under a license agreement, and may be used or copied only in accordance with the terms thereof. It is against the law to copy the software on any other medium, except as specifically provided in the license agreement. The licensee may make one copy of the software for backup purposes. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopied, recorded or otherwise, without the prior written permission of PortaOne Inc.

The software license and limited warranty for the accompanying products are set forth in the information packet supplied with the product, and are incorporated herein by this reference. If you cannot locate the software license, contact your PortaOne representative for a copy.

All product names mentioned in this manual are for identification purposes only, and are either trademarks or registered trademarks of their respective owners.

## Table of Contents

Preface .....	4
What is new in Maintenance Release 75? .....	5
<b>1. PortaBilling RADIUS interface .....</b>	<b>6</b>
RADIUS protocol .....	7
RADIUS attributes.....	7
<i>Authentication / Authorization requests.....</i>	<i>7</i>
<i>Authentication / Authorization responses.....</i>	<i>8</i>
<i>Accounting requests.....</i>	<i>9</i>
RADIUS attributes for Internet billing.....	10
<i>Authentication / Authorization requests.....</i>	<i>11</i>
<i>Authentication / Authorization responses.....</i>	<i>11</i>
<i>Accounting requests.....</i>	<i>11</i>
PortaBilling® custom attributes .....	12
<i>Attributes sent by a RADIUS client (gateway) to the PortaBilling®</i>	<i>server.....</i>
.....	13
<i>Attributes sent by the PortaBilling® server to a RADIUS client</i>	<i>(gateway).....</i>
.....	26
RADIUS return codes .....	33
<b>2. PortaBilling Diameter interface .....</b>	<b>35</b>
Diameter protocol.....	36
<i>Interfaces.....</i>	<i>36</i>
<b>3. External system provisioning framework.....</b>	<b>37</b>
Overview.....	38
<i>Provisioning via event handlers.....</i>	<i>39</i>
<i>Provisioning via webhooks.....</i>	<i>40</i>
<i>Supported event handlers.....</i>	<i>42</i>
<i>Supported event types.....</i>	<i>45</i>
<i>Filters.....</i>	<i>53</i>
<b>4. Appendices.....</b>	<b>56</b>
Appendix A. Examples of RADIUS communication.....	57
<i>Failed login (Authentication failure).....</i>	<i>57</i>
<i>Failed outgoing call (Authorization failure).....</i>	<i>57</i>
<i>Successful prepaid card IVR session.....</i>	<i>58</i>
<i>SIP routing request.....</i>	<i>59</i>
<i>Callback session.....</i>	<i>60</i>
<i>Voucher recharge session.....</i>	<i>62</i>
<i>Music download service requests.....</i>	<i>62</i>
<i>Internet session.....</i>	<i>63</i>
<i>xDR import post-processing.....</i>	<i>65</i>
Appendix B. Examples of SIP communication .....	66
<i>Messaging via SIP SIMPLE.....</i>	<i>66</i>
Appendix C. Description of xDR record fields .....	70

## Preface

This document provides information for developers who want to interface their applications, platforms or systems with PortaSwitch®. The XML / JSON API interfaces are described in the [PortaBilling XML / JSON API Reference](#) and [PortaSIP XML / JSON API Reference](#).

### Where to get the latest version of this guide

The hard copy of this guide is updated upon major releases only, and does not always contain the latest material on enhancements that occur in-between minor releases. The online copy of this guide is always up to date, and integrates the latest changes to the product. You can access the latest copy of this guide at: [www.portaone.com/support/documentation/](http://www.portaone.com/support/documentation/).

## Conventions

This publication uses the following conventions:

- Commands and keywords are given in **boldface**.
- Terminal sessions, console screens, or system file names are displayed in `fixed width font`.



**Exclamation mark** draws your attention to important actions that must be taken for proper configuration.

**NOTE:** Notes contain additional information to supplement or accentuate important points in the text.



**Timesaver** means that you can save time by taking the action described here.



**Tips** provide information that might help you solve a problem.



**Gear** points out that this feature must be enabled on the Configuration server.

## Trademarks and copyrights

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

---

## What is new in Maintenance Release 75?

### Updated:

- The [Supported external systems](#) chapter.
- The [Supported event types](#) chapter.
- The [Filters](#) chapter.

# 1 ■ PortaBilling RADIUS interface

## RADIUS protocol

PortaBilling® uses the RADIUS protocol as per **Remote Authentication Dial-In User Service (RADIUS), RFC 2865** and **RADIUS Accounting, RFC 2866**. By default, the PortaBilling® RADIUS server listens on port 1812 (UDP) for authentication requests and on port 1813 (UDP) for accounting requests.

### RADIUS attributes

PortaBilling® uses a set of Cisco RADIUS VSAs (Vendor-Specific Attributes). For more information, please refer to the RADIUS Vendor-Specific Attributes Voice Implementation Guide at **Cisco website** (<http://www.cisco.com>).

## RADIUS attributes

To ensure compatibility with future releases of PortaBilling®, it is highly recommended that you follow the Cisco guidelines regarding which attributes must be contained in each authentication or accounting request. For more information, please refer to the RADIUS Vendor-Specific Attributes Voice Implementation Guide at **Cisco website** (<http://www.cisco.com>).

### Authentication / Authorization requests

Currently, the following attributes are required for correct processing of authentication / authorization requests:

Attribute	Description
NAS-IP-Address (required)	Specifies the IP address of the network access server that is requesting authentication
NAS-Identifier	Specifies the name (e.g. the hostname) of the network access server that is requesting authentication. Either NAS-IP-Address or NAS-Identifier attributes must be present in the authentication request
User-Name (required)	Indicates the name of the user being authenticated by the RADIUS server
Password	Indicates the user's password
Called-Station-Id	The telephone number the user called. <b>Only for authorization requests</b>
Message-Authenticator	Used to authenticate and protect the integrity of Access-Requests in order to prevent spoofing

h323-conf-id	The unique call identifier generated by the gateway. Used to identify the separate billable events (calls) within a single calling session
h323-incoming-conf-id	<p>The unique number for identifying a parent session on a gateway, where a session is closed when the calling party hangs up. It is used to:</p> <ul style="list-style-type: none"> <li>• Match the outbound and inbound call legs for a session on a particular gateway;</li> <li>• Collect and match all records for multiple calls placed (within the bounds of a session) on the gateway.</li> </ul> <p>The value used for legs 1 and 2 on the originating gateway can differ from that for legs 3 and 4 on a terminating gateway. The h323-incoming-conf-id is different from h323-conf-id.</p>
h323-ivr-out	User-definable attribute-value pairs sent from the NAS to the RADIUS server
PortaOne-*	PortaOne specific attributes sent from the NAS to the RADIUS server. The list of attributes is described in the PortaBilling® custom attributes section

## Authentication / Authorization responses

Currently, the following standard attributes are used (A1 = authentication, A2 = authorization):

Attribute	A1	A2	Description
h323-billing-model	Y		Type of billing service for a specific session (0 Credit, 1 Debit)
h323-return-code	Y	Y	Return codes are the server's instructions to the network access server. The list of possible values are described in the <i>RADIUS Return Codes</i> section
h323-currency	Y	Y	Currency used with h323-credit-amount (in ISO_4217 format, for example USD, EUR)
h323-preferred-lang	Y	Y	Language to use for the audio prompt (in ISO 639-1 format, for example en, es, de)
h323-credit-amount	Y		Amount of credit (in currency) in the account
h323-redirect-number	Y		Optional



			Phone number to which the call is redirected
h323-ivr-in	Y	Y	User-definable attribute-value pairs sent from the RADIUS server back to the NAS
Message-Authenticator	Y	Y	Used to authenticate and protect the integrity of Access-Requests in order to prevent spoofing.
Reply-Message	Y	Y	The description of the accept or reject message
PortaOne-*	Y	Y	PortaOne specific attributes sent from the RADIUS server to the NAS. The list of attributes is described in the PortaBilling® custom attributes section

## Accounting requests

Currently, the following attributes are required for correct processing of accounting requests:

Attribute	Description
NAS-IP-Address	Specifies the IP address of the network access server that is sending accounting requests
NAS-Identifier	Specifies the name (e.g. the hostname) of the network access server that is requesting authentication. Either NAS-IP-Address or NAS-Identifier attributes must be present in the request
NAS-Port-Name	Indicates the name of the physical NAS port that is authenticating the user
User-Name	Indicates the name of the user for the accounting request
Calling-Station-Id	The telephone number the call or message came from
Called-Station-Id	The telephone number the user called / sent a message
Acct-Session-Time	Indicates how long (in seconds) the user has received service
Acct-Status-Type	Indicates the stage of the session, whether this Accounting-Request marks the beginning of the user service (Start), interim update (Alive) or the end (Stop)
h323-remote-address	IP address of the remote voice equipment. PortaBilling® uses it to disambiguate different IP route paths on a NAS

h323-setup-time	Indicates the setup time in Network Time Protocol (NTP) format: hour, minutes, seconds, microseconds, time_zone, day, month, day_of_month, year
h323-connect-time	Indicates the connect time in Network Time Protocol (NTP) format: hour, minutes, seconds, microseconds, time_zone, day, month, day_of_month, and year
h323-disconnect-time	Indicates the disconnect time in Network Time Protocol (NTP) format: hour, minutes, seconds, microseconds, time_zone, day, month, day_of_month, year
h323-conf-id	The unique call identifier generated by the gateway. Used to identify the separate billable events (calls) within a single session. When this attribute is not present, PortaBilling calculates the equivalent ID from other attributes, for example Acct-Session-Id
h323-incoming-conf-id	The unique number for identifying a calling session on a NAS, where a session is closed when the calling party hangs up. The h323-incoming-conf-id number is used to: <ul style="list-style-type: none"> <li>• Match the outbound and inbound call legs for a session on a particular NAS</li> <li>• Collect and match all records for multiple calls placed (within the bounds of a session) on the gateway</li> </ul>
h323-disconnect-cause	Specifies the reason a call was disconnected (e.g. 0x10 - normal ending, 0x3 no route, 0x11 user busy, 0x1F normal unspecified)
h323-call-origin	The NAS'es behavior in relation to the connection that is active for this leg. Possible values: answer, originate. For example, answer on leg 1; originate on leg 2
h323-call-type	Protocol type or family used on this leg of the call. Possible values: VoIP, Telephony
h323-ivr-out	User-definable attribute-value (AV) pairs sent from the NAS to the RADIUS server

## RADIUS attributes for Internet billing

An Internet session consists of the following RADIUS requests:

- Authorize::NETACCESS
- Accounting::NETACCESS::Start

- Accounting::NETACCESS::Alive
- Accounting::NETACCESS::Stop

## Authentication / Authorization requests

Currently, the following attributes are required for the correct processing of authentication / authorization requests:

Attribute	Description
User-Name (required)	This is the name of the user being authenticated by the RADIUS server.
Password	Encrypted password with PAP or CHAP authentication. Present only in an authentication request.
NAS-IP-Address (required)	Specifies the IP address of the network access server that is requesting authentication.
NAS-Port-Type	Indicates the type of the port on the network access server that is requesting authentication.
NAS-Port	Indicates the port on the network access server that is requesting authentication.
Service-Type	The type of service the user is requesting.
Framed-IP-Address	The IP address to assign to a user. It can be defined for a user under an account configuration or allocated by NAS from a pool of IP addresses.

## Authentication / Authorization responses

The billing server can generate one of three responses to NAS:

- **Access Reject** – The user is unconditionally denied access to all requested network resources. Reasons may include failure to provide proof of identification or unknown or inactive user account.
- **Access Challenge** – Additional information such as a secondary password, PIN, etc. is requested from the user. Access Challenge is also used in complex authentication dialogues.
- **Access Accept** – The user is granted access. Once the user is authenticated, the billing server will check that the user is authorized to use the network service requested.

## Accounting requests

The focus of accounting is to track both network resources usage and traffic characteristics.

The following attributes are used for the correct processing of accounting requests:

Attribute	Description
Acct-Status-Type	The type of accounting message: Start, Alive, Stop.
Calling-Station-Id	The user's telephone number for a Dialup session, MAC address for a PPPoE session and the IP address for a PPP session.
Called-Station-Id	The telephone number the user calls for a Dialup session, MAC address for a PPPoE session and the user IP address for a PPP session.
Acct-Session-Time	Defines how long the user is connected (for Alive requests) or was connected (for Stop requests).
Acct-Session-Id	A unique identifier for the session.
Acct-Input-Octets	The user's outgoing traffic (in bytes).
Acct-Output-Octets	The user's incoming traffic (in bytes).
Acct-Input-Packets	Indicates the number of received packets.
Acct-Output-Packets	Indicates the number of sent packets.
User-Name	Indicates the name of the user for the accounting request.
NAS-IP-Address	Specifies the IP address of the network access server that is sending accounting requests

## PortaBilling® custom attributes

All custom attributes conform to the VSA syntax. Attributes received by PortaBilling® from the NAS are entered into the `h323-ivr-out` attribute (VSA No. 1), encoded as `name:value` pairs. Attributes sent by PortaBilling® to the NAS are entered into the `h323-ivr-in` attribute (VSA No. 1), encoded as `name:value` pairs. The second form e.g. `h323-ivr-out = PortaBilling_Session:value` is used to simplify integration with historical third party equipment already supporting Cisco VSA dictionaries.

Unless otherwise specified, any custom attribute may only be included once for each request.

## Attributes sent by a RADIUS client (gateway) to the PortaBilling® server

### PortaBilling\_AuthMethod

This is the internal attribute used to describe the method of authentication or authorization requested by NAS application from the billing server. Normally the method is determined from service type and context automatically, but advanced applications (e.g. IVR) may override it. The value is used in billing logs to differentiate among request types.

Format:

PortaBilling\_AuthMethod:<string>

Possible values:

INVITE, REFER, REGISTER for various SIP voice call authorizations

Example:

h323-ivr-out = 'PortaBilling\_AuthMethod:INVITE'

### PortaBilling\_Authorize

This attribute is used by NAS to request a quota during authorization or reauthorization from the billing server. It is then returned to the NAS in the authorization response from the billing server with provided properties of a granted quota.

Format:

PortaBilling\_AuthMethod:base=<rating\_base>;<option1>=<value1>;<option2>=<value2>;...

Possible values for the authorization request:

Option	Description
<b>base</b>	The rating base which the application can support for maintaining usage counters, for example session-time. In general, the application can support multiple base counters, so that the attribute can be presented multiple times
<b>min</b>	The minimum amount of units (>0) for this rating base till the next authorization. This value also informs the billing engine that the NAS is capable of sending re-authorization requests
<b>max</b>	The maximum amount of units (>0) for this rating base which the application is requesting. This represents the resource limit for authorizing this session
<b>max_funds</b>	The maximum amount of funds to be allocated for the session. This is optional, as fund limits are usually either

	set on the PortaBilling side, or computed dynamically based on the price and value provided in <i>max</i> option
<b>expires</b>	UNIX timestamp which defines the moment when the transaction is expected to start (for example, covering the delay before a call is established). Optional; defaults to authorization request time
<b>max_expires</b>	<p>An estimate of when the maximum amount of units could be spent, in order to properly estimate the time for fund lock expiry. This helps to compute the speed (in seconds) at which the units are spent. It is only needed if the base units are not in seconds.</p> <p>For session-time, the billing engine can compute this as <math>\text{expires} + \text{max}</math>, with the spending speed as 1 unit per second. In general, the spending speed can be computed as <math>\text{max} / (\text{max\_expires} - \text{expires})</math></p>

Example:

```
h323-ivr-out = 'PortaBilling_Authorize:base=session-
time;max=10;min=5'
```

Possible values for the authorization response:

Option	Description
<b>base</b>	The rating base which the NAS should use to limit the resources
<b>avail</b>	The maximum amount of resource units available for the session. This can be omitted if there is another standard attribute for maximum session units provided (like h323-ivr-out=DURATION:); otherwise the default for this value is -1. If the value is negative, it means “infinite”, and the session resources are not limited
<b>expires</b>	UNIX timestamp informing NAS of the moment when the relevant deposit lock will expire
<b>retained</b>	When the resource limit has been used, this number shows how many units were retained (not provided for this session) due to the limit

Example:

```
h323-ivr-in = 'PortaBilling_Authorize:base=session-
time;avail=10;retained=490;expires=1515660408'
```

### PortaBilling\_AccessCode

An access code for selecting a specific rate plan within a product. Although originally used to pass the “IVR access number” to billing, it may contain any string (e.g. “FIRSTCALL”). For Quantum compatibility, this attribute also has the synonym ACCESSCODE.

Format:

PortaBilling\_AccessCode:<string>

Possible values:

Any string. The value passed will be compared to the values entered in the product's accessibility (case-sensitively).

1. **Exact match** – Any ASCII characters string without '|' and '.' equal to the PortaBilling access code to be matched (case-sensitively).
2. **Dot separated match pattern** – The string which consists of the "domain" like string separated by dots (e.g. 'ivr.branding.4565'). It is split on dots and matched by the longest split, e.g. to match 'ivr.branding.4565' pattern against 'ivr.branding.4565', 'ivr.branding' and 'ivr'.
3. **The '|' separated list of patterns** – the dot separated patterns are tried from the left and the first matching one will win. E.g. 'top.321|ivr.branding' matches in the order: 'top.321', 'top', 'ivr.branding', 'ivr'.

Example:

h323-ivr-out = 'PortaBilling\_AccessCode:18001234567'

### PortaBilling\_CallbackHistory

This provides information about the other (second) call leg of a call in the callback scenario, so that the first call leg can be billed properly (e.g. “do not bill the first leg if the second leg was not connected”).

This attribute may be present multiple times in a request. Only the last occurrence is actually used, while the others are ignored.

Format:

PortaBilling\_CallbackHistory:<string>

Possible values:

Value	Description
START	The first call leg was started
ATTEMPT	The callback engine attempted to establish an outgoing call for the second leg
OK	The second call leg was successfully connected

If there is no OK entry, the call leg will be billed using a special rate plan associated in the product configuration with the “CALLBACK\_FAIL” access code.

Example:

h323-ivr-out = 'PortaBilling\_CallbackHistory:START'  
h323-ivr-out = 'PortaBilling\_CallbackHistory:ATTEMPT'  
h323-ivr-out = 'PortaBilling\_CallbackHistory:OK'

### PortaBilling\_Ignore\_Password

This allows to bypass the password check during authentication or authorization.

Format:

PortaBilling\_Ignore\_Password:<YES|NO>

Possible values:

YES or NO.

Example:

h323-ivr-out = PortaBilling\_Ignore\_Password:YES'

### PortaBilling\_Original\_DNIS

This replaces the value of the Called-Station-Id attribute (i.e. the destination phone number used for call rating) to workaround specific call scenario when it cannot be set directly (e.g. when Cisco TCL scripts are used).

Format:

PortaBilling\_Original\_DNIS:<value>

where the value is a semicolon-separated list of name:value pairs.

Possible values:

E.164 phone number (a sequence of digits, \* and/or # signs).

Example:

h323-ivr-out = 'PortaBilling\_Original\_DNIS:19001234567'

### PortaBilling\_RestoreCacheLevel

This controls how the billing engine uses the account information retrieved during the authorization process for a request, in order to facilitate the processing of other requests. Its most common application is to disable this information for re-use, if for some reason information from the current request should not be applied to other call legs (e.g. in callback scripts).

Format:

PortaBilling\_RestoreCacheLevel:<integer>

Possible values:

Value	Description
-1	Disable caching and reuse of account information.
0	Trust the account identified on the request, but not the billing access context



1	Trust both - the account and the billing access context associated with the request
---	---

Example:

```
h323-ivr-out = 'PortaBilling_RestoreCacheLevel:-1'
```

### PortaBilling\_SelectCode

This contains a selection code dialed by a user to route a call via a particular routing plan. PortaBilling® matches the tariff within the product based on the routing plan selection.

Format:

```
PortaBilling_SelectCode:<value>
```

Possible values:

An integer identifying a routing plan select code (prefix).

Example:

```
h323-ivr-out = 'PortaBilling_SelectCode:11564'
```

### PortaBilling\_Session

This allows you to control an account's login session (sessions are used to prevent simultaneous logins by debit accounts). This attribute may be used in the accounting record for a specific call leg in order to force the session to be unlocked. (For instance, to force a session unlock on the answer/VoIP call leg, instead of the customary answer/Telephony call leg). An optional `expires` parameter may control how long the session lock state is supposed to be held on the billing engine (for example, to preserve the funds deposited for the session until the next radius message.)

Format:

```
PortaBilling_Session:<string>
```

```
PortaBilling_Session:lock=<lockvalue>[;expires=<seconds>]
```

Possible lock values:

Value	Description
nolock	Do not lock the account after the current request
unlock	Do not lock the account for the current session and unlock any other existing session for this account
relock	Lock the account for the current session and unlock any other existing session for this account
ignore	Do not set or remove any locks, and skip any lock checks

Example:

```
h323-ivr-out = 'PortaBilling_Session:nolock'
```

### PortaBilling\_SessionEnv

This allows specifying the value of `i_env` for request from internal NAS node. For normal nodes with already defined `i_env` value, this attribute will be ignored.

Format:

`PortaBilling_SessionEnv:<int>`

Example:

`h323-ivr-out = 'PortaBilling_SessionEnv:3'`

### PortaOne-Access-Info

Provides the information about the access points in the authorization request from NAS. An access point identifies the way in which a user accesses services and how are they going to be charged. Multiple services can be allowed for a user in a product (e.g. ANI authentication, wholesale or callback); however, they must be clearly separated from each other to ensure correct billing. This is done by defining a rating entry for every service allowed in a product. PortaBilling uses this attribute to match the rating entry in the product configuration when authorizing a session.

Format:

`PortaOne-Access-Info:<propertyname1=value1>;...`

Example:

`PortaOne-Access-Info = 'delegate=1;nas-ip-address=10.10.10.10;user-name=14567238;access-code=OUTGOING'`

### PortaOne-Service-Type (formerly Service-Identifier)

This contains the identifier for a service in PortaBilling®.

Format:

`PortaOne-Service-Type=<value>`

Possible values:

A non-empty string identifying a service in PortaBilling®:

- Session – Generic time-based service type; it can be used to apply charges for any service use based on the length of time the service was accessed.
- Voice – Rating telephony calls (incoming or outgoing) made via PortaSIP®, VoIP gateways or other equipment.
- Data – Data transfers rated using the amount transferred as the billing parameter.
- Netaccess – Internet access sessions (DSL, PPPoE, etc.), rated based on session duration or the amount of transferred data.

- Quantity – Generic quantity-based service type; can be used to apply charges for any service use expressible in numerical form (e.g. the number of pizzas ordered).
- Wifi – Wireless Internet access sessions, rated based on session duration.
- Dialup – Dialup Internet access sessions, rated based on session duration.
- Msg – Rating messages (text, SMS, MMS, other) based on the number of messages sent.
- Conference – Rating conference calls via PortaSIP® Media Server (or some conferencing server).
- Did – Can be used to apply charges to customers for usage of DID numbers.

Example:

PortaOne-Service-Type=Conference

### PortaOne-Called-Party

This provides information about the called party.

Format:

PortaOne-called-Party:<propertyname1=value1>;...

Properties:

Property	Description
id	This contains the caller ID information for the call. It is taken from the <i>P-Asserted-Identity</i> and <i>RPID</i> headers
display-id	This indicates the “Caller number” value that is taken from the From: header and typically displayed on the called party’s phone display
privacy	This indicates that the calling party requests its identification to be kept private (hidden from the final call recipient.) Possible values: 1/0, default 0
name	This is a caller name. It is taken from the CLN field
site	The unique ID of the local Centrex site which the party belongs to
x-id	The caller’s extension number configured on the PBX of a calling party / within the IP Centrex
x	The unique ID of the local IP Centrex (customer/head office) which the party belongs to
ctx-id	The unique identifier for the calling party’s PBX
x-grp	Optional. The group setting for the call group pickup. The huntgroup extension, on behalf of which the call happened
media-time	The time the respective party spent connected to media on the call

bill-to	The account ID representing the entity in system. If not present, it means the party is from the outside network. The value can be used in User-Name if the respective party performs some action (e.g. becomes transferor).
net	The identifier of the network: <ul style="list-style-type: none"> <li>• 0 for directly owned accounts.</li> <li>• &gt;0 is an id of reseller.</li> </ul>
e212	Code in MCC-MNC format (E.212) - mobile operator + network codes
roam-e212	The roaming identification of a caller in the MCC-MNC format (E.212)

Example:

```
h323-ivr-out = 'PortaOne-Called-Party = 'ctx-
id=52333;name=EASYCALL;id=17289277770;display-
id=17289277770;net=0'
```

### PortaOne-Calling-Party

This provides information about the calling party.

Format:

```
PortaOne-Calling-Party = <propertyname1=value1>;...
```

Properties:

Property	Description
id	This contains the caller ID information for the call. It is taken from the <i>P-Asserted-Identity</i> and <i>RPID</i> headers
display-id	This indicates the “Caller number” value that is taken from the From: header and typically displayed on the called party’s phone display
privacy	This is a privacy flag which indicates that the calling party requests its identification to be kept private (hidden from the final call recipient.) Possible values: 1/0, default 0
name	This is a caller name. It is taken from the CLN field
site	The unique ID of the local Centrex site which the party belongs to
x-id	The caller’s extension number configured on the PBX of a calling party
x-grp	The huntgroup extension, on behalf of which the call has happened
x	The unique identifier for the calling party PBX
media-time	The time the respective party spent connected to media on the call
bill-to	The account ID representing the entity in system. If not present, it means the party is from the outside network. The value can be used in User-Name if the respective party performs some action (e.g. becomes transferor).

net	The identifier of the network: <ul style="list-style-type: none"> <li>• 0 for directly owned accounts.</li> <li>• &gt;0 is an id of reseller.</li> </ul>
ctx-id	The unique identifier for the calling party's PBX
e212	Code in MCC-MNC format (E.212) - mobile operator + network codes
roam-e212	The roaming identification of a caller in the MCC-MNC format (E.212)

Example:

```
h323-ivr-out      = 'PortaOne-Calling-Party =
'x=52333;name=EASYCALL;id=17289277770;display-id=17289277770''
```

### PortaOne-Charge

This informs that the imported xDR must be charged.

Format:

```
PortaOne-Charge:<propertyname1=value1>;...
```

Properties:

Property	Description
apply	The request flag which defines when the charge is applicable for a certain entity-type. Possible values: <ul style="list-style-type: none"> <li>• always - apply the charge regardless of context;</li> <li>• connection - the charge is applicable when a connection is matched;</li> <li>• never - the charge is not applied</li> </ul>
commit	The request flag which defines the condition on which the charges are to be committed to the database. Possible values: <ul style="list-style-type: none"> <li>• always (default) - commit all created charges;</li> <li>• complete - the charges are only stored/committed if all charges relevant for the event are applied;</li> <li>• never - the charges are not stored at all. We recommend using it only for testing the "dry run" charging</li> </ul>
report	The request flag to return charge results in the response
type	The type of charging. Possible values:

	<ul style="list-style-type: none"> <li>charge – to add additional charge amount</li> <li>prerated – to additionally store it into Aux record (used only for requests)</li> </ul>
entity-type	The type of the charged entity. Possible values: account/reseller/reseller[level]/vendor
storage-type	The type of the backend storage where this xDR record is added: Account/AccountFail, Customer/CustomerFail, Vendor/VendorFail, OnNet
charged-amount	The value of <i>xdr.charged_amount</i>
charged-quantity	The value of <i>xdr.charged_quantity</i> (in units/rating base according to the service)
i_account/i_customer/i_vendor	Identifier of the entity being charged
flag.*	The indicators associated with the charge. For example, flag.block=<threshold> informs that usage already passed into the discount stage, where the service is blocked and that it happened on a threshold with the given value

**Example:**

```
PortaOne-Charge = 'entity-type=account;apply=always'
PortaOne-Charge = 'type=charge;entity-type=account;charged-amount=3.2;currency=USD;history=prerated'
```

**PortaOne-Redirecting-Party**

This provides information about the redirecting party in case the call is forwarded or transferred.

**Format:**

```
PortaOne-Redirecting-Party:<propertyname1=value1>;...
```

The properties and values are the same as for the *PortaOne-Calling-Party* attribute described above.

**PortaOne-Media-Info**

This provides information about the consumed bandwidth according to the formula reflecting the connectivity and media.

**Format:**

```
PortaOne-Media-Info:<propertyname1=value1>;...
```

Properties:

Property	Description
direction	Direction of the RTP stream on the leg relative to proxy
type	The mime type style description of the codec
ptime	The packetization time in milliseconds, which determines the amount of audio data included in a single packet
bitrate	The selected bitrate in bits per second
proxy	Indication of whether the RTP proxy was used. 0 - default

Example:

```
PortaOne-Media-Info =
'bitrate=64000;direction=in;type=audio/PCMU;proxy=1;ptime=20'
```

### PortaOne-Geo-Location

This provides information about a user's geographical location when they use a service. The type of identification or it's precision may vary based on the service scenario

Format:

PortaOne-Geo-Location:<the geo location specifier in dot notation starting with iso country code>

Example:

```
PortaOne-Geo-Location:fr.Paris
```

### PortaOne-Service-Features

This attribute is used by NAS to request the service features (capabilities) which are supported for the service from the billing engine. The billing engine returns the indications of service features to be provided in the response.

Format:

PortaOne-Service-Feature:<propertyname1=value1>;...

The list of service features which can be requested / declared:

Name	Description
incoming-charge	The option to authorize incoming call to destination party as a separate event (this is the default capability of PortaSIP)
x-nas-ip-address	Reference to the Centrex (on-net managing) node to determine the context for IP Centrex features in case another node is requesting the features

x-nas-identifier	Reference to the Centrex (on-net managing) node to determine the context for IP Centrex features in case another node is requesting the features
aaa	The currently applied authentication, authorization and accounting mode, which defining what requests/events are sent to the billing engine
app-type	The type of the application (IVR) which requests the service features
routing	The indication to request routing and provide the routing results
sip	The indication to provide routes via the SIP transport protocol (NAS is capable of SIP routing)
h323	The indication to provide routes via the H323 transport protocol (NAS is capable of H323 routing)
direct-only	Indicates to bypass any routes which require a protocol conversion to be performed via converting node
proclaim-duration	NAS is capable of informing about available call duration separately from actual call duration limit (so tricky rating may be used)
bill-to	NAS is capable of replacing User-Name/billed entity on further authorization
translate-cld	NAS is capable of replacing CLD during authorization
pattern	NAS is capable of obtaining and delivering the rate pattern
max-depth	Defines the maximum number of hops allowed for the call to be redirected before reaching the destination
force-cleanup	Forces to immediately cleanup/charge on accounting request receipt. This is the indication that there is no point to cache and wait for other call info
local-lookup	Instructs the billing engine to search for the destination among local accounts
e212	Indicates that the destination is in the e212 format and that NAS is capable of / to process e212 destination in rating/routing
e164	Indicates that the destination is in the e164 format and that NAS is capable of / to process e164 destination in rating/routing
negotiable-mask	The mask of media and other supported features available for connecting
criteria	Indicates the capability of the NAS to process routing with the consideration of adaptive routing criteria



utilization	Indicates the capability of the NAS to rearrange routes in a list to according to the connection utilization parameter
points-required	Requires a routing response to include valid routes, otherwise the call should be rejected
charge-report	Requests accounting response with charge report filled in
commit-charge	Commit the charge to the database (default). It can be set to 0 to bypass commit to perform charging in a test mode

Example:

```
PortaOne-Service-Features = 'routing=1;'
```

### PortaOne-Service-Action

This allows NAS to perform special actions when BE responds with the authorization.

Format:

```
PortaOne-Service-Action:action=<action type>;param1=<param1 value>;...
```

Example:

```
PortaOne-Service-Action = 'action=call-pickup;id=call-pickup;'
```

### PortaOne-Transport-Protocol

This shows the transport protocol. Type can be one of the following:

- SIP
- H323
- SMS
- SMPP

Format:

```
PortaOne-Transport-Protocol = <type>
```

Example:

```
h323-ivr-out          ='PortaOne-Transport-Protocol = 'SIP''
```

### PortaOne-Leg-Type

This shows whether the request is originate or answer. Type can be one of the following:

- 1 – Answer request
- 2 – Originate request

Format:

```
PortaOne-Leg-Type:<type>
```

Example:  
 h323-ivr-out = PortaOne-Leg-Type = '1'

### PortaOne-Leg-Origin

This shows the leg origin.

Format:  
 PortaOne-Leg-Origin:<type>

Example:  
 h323-ivr-out = PortaOne-Leg-Origin = '0'

### PortaOne-Used-Resource

This specifies the amount of used service units measured from the point when the service session identified in the h323\_conf\_id attribute becomes active.

Format:  
 PortaOne-Used-Resource:base=<base>;amount=<value>

Possible values:  
 <base> – One of: quantity, session-time  
 <value> – Integer number specifying amount of measurement units.

Example:  
 PortaOne-Used-Resource = base=quantity;value=2  
 PortaOne-Used-Resource = base=session-time;value=70

## Attributes sent by the PortaBilling® server to a RADIUS client (gateway)

Attribute	A1	A2	Description
DURATION		Y	The maximum allowed call duration (in seconds)
Tariff	Y	Y	The name of the rate plan applied to the account within this session
available-funds	Y		For debit accounts, this is equal to the h323-credit-amount. For credit accounts, this returns the actual amount of available funds (the difference between the credit limit and the current balance with respect to the credit

			limits for the individual account and customer)
ErrorExplanation		Y	A rejection explanation string code, which can be used to identify a corresponding prompt or a message to play to an end user.
h323-preferred-lang	Y	Y	Language preference for the audio prompt
h323-credit-amount	Y		Amount of credit (in currency) in the account
h323-credit-time		Y	Amount of seconds for which the call is authorized
PortaBilling_UserName	Y	Y	The account ID to be used for billing this call
PortaBilling_RatePattern		Y	The rate pattern is a string used by PortaBilling® to match the corresponding rate code/destination during authorization of an account, instead of the Called-Station-Id
PortaBilling_AccessCode	Y	Y	Access code to be used for matching rating entry and tariff
PortaBilling_AccountBalance	Y		The current amount of the account's balance in PortaBilling®
PortaBilling_CustomerBalance	Y		The current amount of the customer's balance in PortaBilling®
PortaBilling_CustomerCreditLimit	Y		The customer's current credit limit in PortaBilling®
PortaBilling_CustomerCreditLimit Threshold	Y		The customer's current balance warning in PortaBilling®
PortaBilling_NodeId	Y		Provides the node identifier to the NAS
BalanceThreshold	Y		Indicates that a customer's balance warning threshold has been reached
PortaBilling_ProductBreakage	Y		The breakage amount of the account's product in PortaBilling®

PortaBilling_No_Disconnect_Warning		Y	Indicates that NAS won't play a "time left" warning (the remaining time announcement in seconds or a "beeping" sound) when a specified number of seconds is left before the call is disconnected (calls will be terminated silently)
Product		Y	Returns the name of the product associated with the user account

## DURATION

This specifies the real allowed maximum call duration (in seconds), which may differ from the announced credit time if billing tricks are applied.

Format:

DURATION:<integer>

Possible values:

An unsigned integer (number of seconds).

Example:

h323-ivr-in = 'DURATION:320'

## Tariff

The name of the rate plan applied to the account within this session.

Format:

Tariff:<string>

Possible values:

Positive integer (number of seconds).

Example:

h323-ivr-in = 'Tariff:ABC prepaid'

## available-funds

This attribute returns the amount of account's available funds without considering any locked funds. For debit accounts, this is equal to the h323-credit-amount. For credit accounts, this returns the actual amount of available funds (the difference between the credit limit and the current balance with respect to the credit limits for the individual account and customer).

**Format:**

available-funds:<decimal>

**Possible values:**

Any positive number formatted with five decimal places; a dot (.) is used as the decimal separator.

**Example:**

h323-ivr-in = 'available-funds:124.78000'

**ErrorExplanation**

This attribute returns a rejection explanation string code, which can be used to identify a corresponding prompt or a message to play to an end user. It is sent in the authentication response within the h323-ivr-in attribute.

**Format:**

ErrorExplanation:<string>

**Possible values:**

An explanation code string according to [RADIUS return codes](#).

**Example:**

h323-ivr-in = "ErrorExplanation:invalid\_account"

**PortaBilling\_UserName**

The account ID to be used for billing this call. The RADIUS client **must** supply this value as the User-Name (or, alternatively, in h323-ivr-out=PortaBilling\_UserName) attribute value in the accounting records for all call legs (incoming and outgoing).

**Format:**

PortaBilling\_UserName:<value>

**Possible values:**

A non-empty string identifying an account in PortaBilling®.

**Example:**

h323-ivr-in = 'PortaBilling\_UserName:16051233355'

**PortaBilling\_RatePattern**

The rate pattern is a string used by PortaBilling® to match the corresponding rate code/destination during authorization of an account, instead of the Called-Station-Id (e.g. when a customer is to be billed according to a special rate such as VOICEONNET, rather than according to the actual number dialed). The RADIUS client **must** supply this value

as the h323-ivr-out=PortaBilling\_RatePattern attribute value in the accounting records for all outgoing legs of this call.

Format:

PortaBilling\_RatePattern :<value>

Possible values:

A non-empty string, usually a destination number in e.164 format.

Example:

```
h323-ivr-in          = 'PortaBilling_RatePattern:
18001234567|VOICE\ONNET\RX
```

### PortaBilling\_AccessCode

The string actually used by PortaBilling® to select a specific rate plan within a product. Although originally used to pass the “IVR access number” to billing, it may contain any string (e.g. “FIRSTCALL”) or it can compose advanced pattern to best match against the access code configured in the product. NAS should copy this value into all future call accounting requests as h323-ivr-out=PortaBilling\_AccessCode, in order to apply the same properties during billing.

Format:

PortaBilling\_AccessCode:<value>

Possible values:

Any string. The value passed will be compared to the values entered in the product’s accessibility (case-sensitively).

1. **Exact match** – Any ASCII characters string without '|' and '.' equal to the PortaBilling access code to be matched (case-sensitively).
2. **Dot separated match pattern** – The string which consists of the "domain" like string separated by dots (e.g. 'ivr.branding.4565'). It is split on dots and matched by the longest split, e.g. to match 'ivr.branding.4565' pattern against 'ivr.branding.4565', 'ivr.branding' and 'ivr'.
3. **The '|' separated list of patterns** – the dot separated patterns are tried from the left and the first matching one will win. E.g. 'top.321|ivr.branding' matches in the order: 'top.321', 'top', 'ivr.branding', 'ivr'.

Example:

```
h323-ivr-in          =
'PortaBilling_AccessCode:18001234567|prepaid.tollfree'
```

### PortaBilling\_AccountBalance

The current amount of the account’s balance in PortaBilling®.

Format:

PortaBilling\_AccountBalance:<value>

Possible values:

A number with a precision of five decimal places.

Example:

h323-ivr-in = 'PortaBilling\_AccountBalance:13.20000'

### PortaBilling\_CustomerBalance

The current amount of the customer's balance in PortaBilling®.

Format:

PortaBilling\_CustomerBalance:<value>

Possible values:

A number with a precision of five decimal places.

Example:

h323-ivr-in = 'PortaBilling\_CurstomerBalance:13.20000'

### PortaBilling\_CustomerCreditLimit

The current amount of the customer's credit limit in PortaBilling®.

Format:

PortaBilling\_CustomerCreditLimit:<value>

Possible values:

A number with a precision of five decimal places.

Example:

h323-ivr-in = 'PortaBilling\_CustomerCreditLimit:13.00000'

### PortaBilling\_CustomerCreditLimitThreshold

The current amount of the customer's credit limit threshold in PortaBilling®. Note that the relative threshold limit is converted into an actual currency amount.

Format:

PortaBilling\_CustomerCreditLimitThreshold:<value>

Possible values:

A number with a precision of five decimal places.

Example:

h323-ivr-in =  
'PortaBilling\_CustomerCreditLimitThreshold:12.00000'

### PortaBilling\_Mod\_<modulename>

This is a special form to exchange parameters of a specific module with PortaBilling®. The exact set of parameters depends on the module.

Format:

PortaBilling\_Mod\_<modulename>:[param1=value1;]...

Possible values:

Module specific

Example:

```
h323-ivr-out =  
"PortaBilling_Mod_callback:username=cb44012235210;access-  
code=1200123;cld=+44012235210;in-progress=45;pattern=  
44012235210"
```

### PortaBilling\_NodeId

Provides the name of the NAS node (e.g. the hostname) used to identify it back to NAS. If no name is defined for NAS, it identifies itself with IP address.

Format:

PortaBilling\_NodeId:<nodeid>

Possible values:

Any valid string

Example:

```
h323-ivr-out = 'PortaBilling_NodeId:nashost.domain.foo'
```

### BalanceThreshold

Indicates that the customer's balance warning threshold has been reached.

Format:

BalanceThreshold:<value>

Possible values:

YES

Example:

```
h323-ivr-in = 'BalanceThreshold:YES'
```

### PortaBilling\_ProductBreakage

The breakage amount of the account's product in PortaBilling®.

Format:

PortaBilling\_ProductBreakage:<value>



Possible values:

A number with a precision of five decimal places.

Example:

```
h323-ivr-in      = 'PortaBilling_ProductBreakage:0.02000'
```

### PortaBilling\_No\_Disconnect\_Warning

Indicates that NAS should cancel IVR/beep warnings before call session timeouts (calls will be terminated silently).

Format:

```
PortaBilling_No_Disconnect_Warning:<value>
```

Possible values:

YES

Example:

```
h323-ivr-in      = 'PortaBilling_No_Disconnect_Warning:YES'
```

### Product

This returns the name of the main product associated with the user's account in PortaBilling.

Format:

```
Product:<value>
```

Example:

```
h323-ivr-in = 'Product:Smartcard'
```

## RADIUS return codes

Return codes can be defined by the user through the implementation of the IVR script and RADIUS server agreement. The following table lists the return codes that are predefined and anticipated by existing Cisco IVR scripts.

Explanation code	h323-return-code	Description
success	0	Success, proceed
invalid_account	1	Failed - Invalid Account number
invalid_password	2	Failed - Invalid Password
account_in_use	3	Failed - Account in use
bandwidth_limit_exceeded	3	Failed - Bandwidth limit exceeded

call_limit_exceeded	3	Failed - Call limit exceeded
zero_balance	4	Failed - Zero balance
card_expired	5	Failed - Card expired
credit_limit	6	Failed - Credit limit
user_denied	7	Failed - User denied
user_suspended	7	Failed - User suspended
not_available	8	Failed - Service not available
cld_blocked	9	Failed - Called number blocked
cld_barred	9	Failed - CLD is barred
retries_exceeded	10	Failed - Number of retries exceeded
invalid_argument	11	Failed - Invalid argument
insufficient_balance	12	Failed - Insufficient balance
toll_free_allowed	13	Toll-free calls allowed
invalid_account	14	Failed - Invalid card number
pickup_invalid_cld	15	Failed - CLD is not allowed for pick up
hairpin_to_pstn	50	Call will be hairpinned back to the Public Switched Telephone Network (PSTN)
redirect	51	Redirect to called party (use redirect number)
redirect_to_cs	52	Redirect to customer Service (use redirect number)
redirect	53	Connect IP leg to redirect IP address (108)
cld_tmp_unavail	109	CLD is a local account which can't receive the call
cld_unassigned	110	CLD is unassigned in local DID inventory
cld_dial_error	111	CLD is not a valid dialed number
preselect_not_set	112	CLD doesn't define any selection code while no default is set
funds_locked	114	Failed - Funds are locked
reject_authorization	115	Reject authorization of already terminated session
spend_limit_exceeded	116	Failed - Account over its current spending budget
funds_locked	117	Failed - Insufficient funds allocated

# 2. PortaBilling Diameter interface

## Diameter protocol

PortaBilling® uses the Diameter protocol as defined by RFC 6733 Diameter Base Protocol and RFC 4006 Diameter Credit-Control Application with additional AVPs defined in **3GPP TS 32.299** (version 12.0.0 Release 12) for real-time authentication, authorization and user accounting in IMS networks. By default, the billing server listens on port TCP 3868 to serve Diameter requests.

### Interfaces

PortaBilling® supports the following Diameter interfaces:

- **Diameter Gy** interface for real-time content-based data service charging. It is based on the 3GPP standards and relies on quota allocation. Both time- and volume-based charging models are supported.
- **Diameter Ro** interface for real-time voice and SMS traffic charging. It supports two credit authorization modes:
  - direct debiting, wherein user funds are immediately deducted in one single operation (e.g. to charge for an SMS) and
  - unit reservation. In this case the exact amount of service units (e.g. seconds) required to provide the service is not known, thus, a segment of units is reserved and periodically updated for these types of sessions.

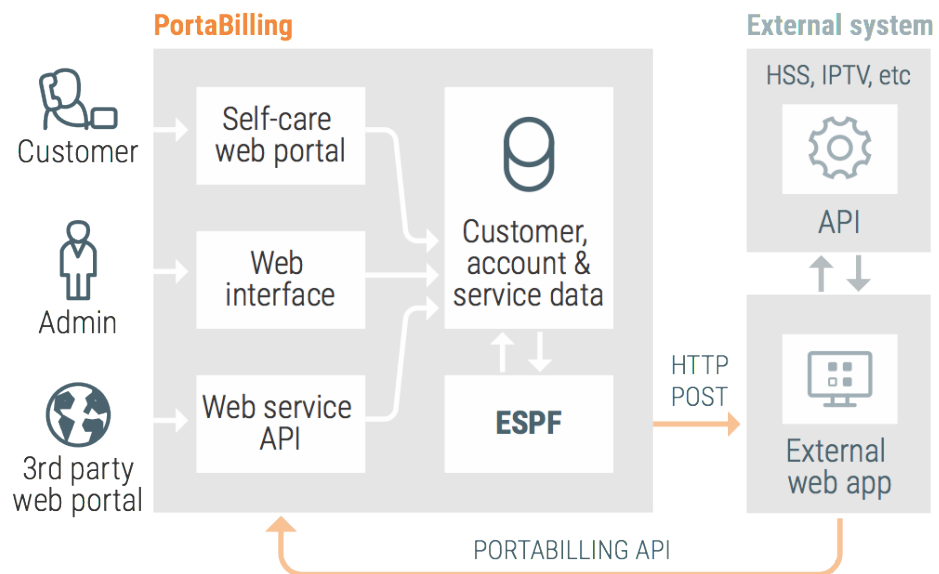
# 3. External system provisioning framework

## Overview

PortaBilling® is a real-time converged B/OSS and OCS system and the range of services you can provide and bill for extends the world of VoIP. For example, you can integrate PortaBilling® with an IPTV platform to provide IPTV services or with a mobile carrier's HSS and PCRF servers to operate as an MVNO and provide LTE services for your users. No matter which external system PortaBilling® integrates with, you must provision data (e.g. SIM card details for HSS) to it and keep all the data synchronized.

This is the job that the external system provisioning framework (ESPF) does. Changes in customers' billing or service configuration that occur in PortaBilling® are recorded as provisioning events. These events are queued in the system. The ESPF captures an event and sends it to the external system to update the data there (e.g. to update the channel package for the subscriber in the IPTV platform if the customer has purchased it via an add-on product in PortaBilling®).

The ESPF simplifies the integration with the external systems that you deploy in your network infrastructure. PortaBilling® provides a single location for data provisioning management.



You can provision data to external systems using:

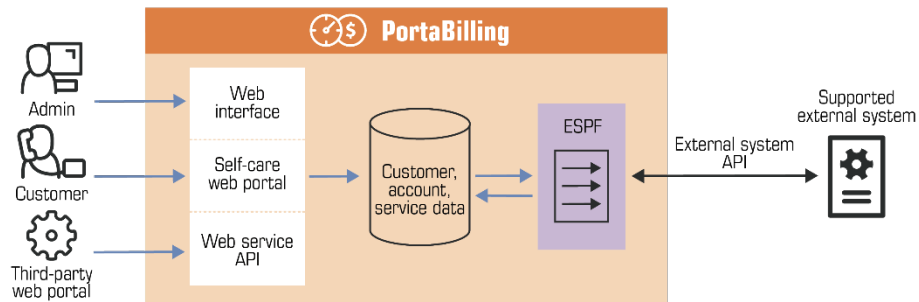
- dedicated event handlers. An event handler is a special Perl module in PortaBilling® that provisions a particular external system using this system's API. There is a separate handler for each external system that PortaBilling® is integrated with. If you need to integrate a new external system and provision data to it,

you can either create your own event handler or request development from PortaOne.

- webhooks. A webhook is an automated notification of an external application about provisioning events that happen in PortaBilling®. You develop an external application that communicates both with PortaBilling® and the external system. When an event occurs (e.g. an account is assigned a SIM card), the ESPF sends the HTTP POST request to the application's URL. The application reads it, retrieves the required information about the entity from PortaBilling® via the API and provisions the data to the external system.

## Provisioning via event handlers

Every event handler provisions a particular external system. You subscribe it to provisioning events that the system requires and the event handler monitors the event queue for these events. When an event occurs, the event handler triggers the API call in the external system to perform the action there (e.g. activate the SIM card).



This is how it works in detail:

1. A database trigger is created for a particular table.
2. The database automatically executes the trigger in response to the defined event and the trigger inserts an event record into the Event\_Queue table. The trigger also inserts a record to the Event\_Notifications table, which stores the event status and the retry counter. This enables you to subscribe several event handlers to provision the same event.
3. The event record contains references to other tables with certain fields that can be provisioned to the external system (e.g. i\_env, ID and balance of an account).
4. A handler subscribed for this event reads the record from the Event\_Queue table and processes it by retrieving the required account or customer information from PortaBilling® via the API. This information differs for each external system (e.g. an HSS requires SIM card details while an IPTV platform requires a channel package name).

5. The handler provisions both the event and its data to the external system and receives a return code (whether the event was processed successfully or not) from it.
6. If the return code is successful, then ESPF changes the event status to “finished” for this handler in the Event\_Notifications table. Otherwise, the handler tries to provision the event again until it is either provisioned or the amount of retries reaches its limit.
7. When all handlers provision this event, the ESPF cleans up the Event\_Notifications table and removes the processed event from the Event\_Queue table.

If an issue occurs during event provisioning, the corresponding error is written to the `provisioning.log` file. In addition, administrators can receive email notifications about provisioning event failures. On the Configuration server, they must specify an email address for the **Provisioning.AlertRecipient** option.

This allows administrators to intercept and resolve event-provisioning issues before they have any significant impact on end users.

### How to configure

To provision a specific event to an external system, follow these steps:

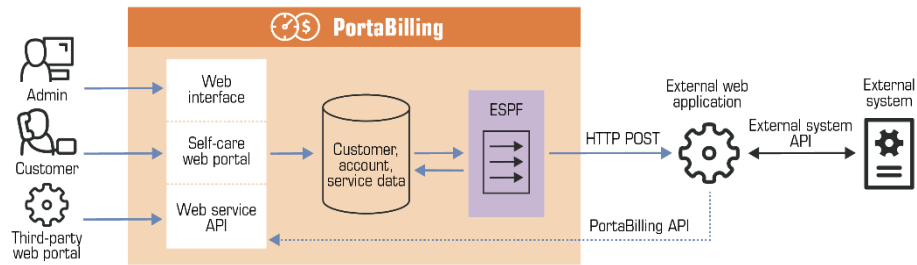
1. Enable the ESPF on the Configuration server web interface.
2. Configure an available event handler to suit your needs on the Configuration server. Find event handlers that are supplied with PortaSwitch® [here](#). Alternatively, you can create your own event handler.
3. Enable the specified events to be triggered. Find the supported events [here](#).
4. Enable the handler that will provide the information to the external system.
5. Subscribe the handler to catch the required events.

Please refer to the **PortaSwitch® Handbook collection** for step-by-step instructions on how to configure the ESPF.

### Provisioning via webhooks

This is an alternative approach to data provisioning. Webhooks allow your external application to receive notifications about provisioning events happening in PortaBilling®.





For simplicity, provisioning events are grouped into four category types:

- **Customer** – The event types of this group notify that a customer record has been created, updated or deleted in PortaBilling®.
- **Subscriber** – The event types of this group notify that an account record has been created, updated or deleted in PortaBilling®.
- **DID** – The event types of this group notify that a DID number has been added, updated (assigned to / unassigned from a customer or account) or deleted in PortaBilling®.
- **Invoice** – The event types of this group notify that an invoice has been created or updated (paid fully or partially, or refunded) in PortaBilling®.

An external application subscribes to receive these event types. Once an event occurs, the ESPF maps it to the corresponding event type and sends the HTTP POST request to the application's URL. The request body contains the event type and the ID of the provisioned entity in JSON format.

For example, when a new SIM card is assigned to a subscriber in PortaBilling®, the ESPF sends an HTTP POST request as follows:

```
http://127.X.X.X:5000:
{
  "event_type": "Subscriber/Updated",
  "variables": {
    "i_account": "1000889"
  }
}
```

Your application receives the request, queries the information related to this event type (SIM card details such as IMSI, MSISDN) from PortaBilling® via the API and triggers the API call in the HSS to activate the SIM card.

## Components

Your external application operates as the middleware between PortaBilling® and the external system it provisions. You can develop it in any programming language (e.g. Java, Python, etc.) and host it on a remote web server or in a cloud.

To provide an additional layer of security to communications between your external system and PortaBilling®, you can use a cryptographic signature for HTTP requests.

The ESPF sends HTTP POST requests using the EventSender handler. It supports all provisioning events and operates in asynchronous mode by default. To prevent system overload in case of large numbers of events, the EventSender controls the number of parallel requests. And, if necessary, you can always change the handler to operate in synchronous mode.

The ESPF can send HTTP POST requests to more than one external application. This enables you to provision data to several external systems at the same time.

For example, you integrate PortaBilling® with the HSS and the IPTV platform to provide both mobile and IPTV services. When you register a new account in PortaBilling®, the ESPF sends a provisioning request to both the HSS and the IPTV platforms.

If a user's service configuration includes only one service (e.g. IPTV), this request is useless for the HSS. Therefore, if a service is not used by a subscriber, make sure that the application to provision the HSS ignores it.

You can configure the ESPF to differentiate among provisioning data and send specific event types to each system. To make this happen, please contact PortaOne support. This configuration requires deep knowledge of ESPF architecture.

## Configuration

Please refer to the [Provisioning Application Reference Guide](#) for guidelines on how to develop an external application to provision data from PortaSwitch®. For guidelines on how to configure the ESPF, please refer to the [PortaSwitch® Handbook collection](#).

## Supported event handlers

### Built-in integrations

PortaBilling® is supplied with these handlers to provision data to external systems:

- **AricentHSS** – This handler provisions subscriber information such as the user's phone number and SIM card details to the Aricent HSS and Adax HSS. It also provisions subscriber's service configuration (static IP address, quality of service information).

- **AudioCodes** – This handler provisions subscriber information such as the user's phone number, SIM card details, status, registration (SIP parameters), mobile device (IMEI, language, etc.) and custom configurations to the AudioCodes CMMS (Clients' Management Server).
- **Bridgewater** – This handler provisions subscriber information such as company / user name, phone number and service configuration (e.g. customer class, product) to Bridgewater, the service controller for WiMAX networks.
- **Calix** – This handler provisions ONT device information such as FSAN serial number plus account and service configuration data (e.g. account status, bandwidth profile ID within the Internet service policy) to the Calix Management System, the platform that controls Internet service availability for subscribers in fiber-optic networks such as GPON / Active Ethernet.
- **EventSender** – This is a universal handler to provision any external system. Since it sends HTTP POST requests with provisioning events to an external application, it's the middleware between PortaSwitch® and an external system. The application can be written in any programming language. It must be able to process provisioning events and communicate with PortaSwitch® and the external system via their API.
- **Huawei::HSS** – This handler provisions subscriber information such as the user's phone number and SIM card details to the Huawei HSS.
- **Huawei::PCRF** – This handler provisions the subscriber's service configuration (service and policy name, QoS parameters) to Huawei PCRF.
- **IPTV** – This handler provisions available channel information to one of the PortaSwitch® supported IPTV platforms (i.e. MatrixStream, Minerva, Telebreeze IPTV) when either new or existing customers sign up for the IPTV service.
- **Jasper** – This handler provisions subscriber and SIM card information to the Cisco Jasper connectivity management platform.
- **Kleos::HSS** – This handler provisions subscriber information such as the user's phone number, their SIM card details and account's status to Greenis HSS.
- **Kleos::PCRF** – This handler provisions the subscriber's service configuration (service and policy name) to Greenis PCRF.
- **LIMADispatcher** – This handler provisions subscriber information such as company / user name, phone number, follow-me number changes to the Lima Platform.
- **NBBS** – This handler provisions user device information to the Motorola NBBS Device Management Platform when a new CPE

is added to the CPE inventory in PortaBilling® and also when a device is assigned to or removed from an account.

- **NetNumber** – This handler provisions ported-in numbers to the NetNumber SMS/MMS database.
- **Protei::HLR** – This handler provisions subscriber information such as the user's phone number, SIM card details and their account's status to Protei HSS (for centralized user authentication across the network).
- **Protei::PCRF** – This handler provisions the subscriber's service configuration to PCRF (to allocate the correct policy / QoS parameters to the customer).
- **ReadyWireless** – This handler provisions the subscriber's phone number and device (IMEI) to ReadyWireless MVNE to trigger device activation / deactivation.
- **Titan::HSS** – This handler provisions subscriber information such as the user's phone number and SIM card details to NetNumber Titan HSS. It also provisions subscriber's service configuration – Regular upload / download limits for Internet access, APN (Access Point Name) ID, Roaming profile ID.
- **YateHSS** – This handler provisions subscriber information such as the user's phone number, their SIM card details and user's status to Yate HSS/HLR.
- **ZXUN::HSS** – This handler provisions subscriber information such as the user's phone number, SIM card details and their account status to the ZTE HSS.
- **ZXUN::SPR** – This handler provisions the subscriber's service configuration, user's phone number and SIM card details to the ZTE SPR.

### Internal handlers

Changes in customer, product and service configuration influence the configuration of accounts in PortaBilling®. For example, when a customer reaches their credit limit, services are no longer available to their credit accounts until the customer's balance is refilled.

To reduce the load on the system in such cases, ESPF is supplied with handlers that process product, customer and service configuration events, and then create events for accounts and place them in the event queue:

- **CustomerToAccountsDispatcher** – This handler processes customer status changes and provisions them to all customer accounts.
- **ProductToAccountsDispatcher** – This handler processes all changes at the product level and provisions them to an account that has this product assigned to it.
- **ServiceAttributeDispatcher** – This handler provisions changes in service configuration to both customers and accounts. You can

specify filters for this event handler so you reduce the number of queries to the database. See the [Filters](#) section for details.

These handlers process events within PortaBilling®:

- **SessionDisconnect** – This handler adds tasks to the specific event queue for further active session disconnection.
- **SIPForwarder** – This handler is used when an account is moved from one system to another. It provisions information about an account and / or the DID number location to the dispatching SBC in the dual-version PortaSwitch® deployment.

These handlers can be used as templates for developing custom handlers:

- **DemoHandler** – This handler is an example of the event handler. Use it to create a custom handler.
- **Sokoban::Log** – This handler writes the data into the log file (e.g. testing, debugging, etc.).

## Supported event types

The ESPF is supplied with the following set of provisioning event types:

Name	Description	Inherited from Customer
Accessibility/Deleted	The usage charges record has been removed from the account's product.	
Accessibility / Inserted	A usage charges record has been added to the accounts product.	
Account/ActivationDate/Changed	The activation date has been changed for an account. (Accounts.activation_date has been changed)	
Account/Alias/Delete	An alias has been removed from an account.	
Account/AvailableFundsAppear	An account's available funds are restored when a debit account tops up the balance or when a postpaid customer pays their invoice. (Accounts.balance has been topped up from credit_limit.) I_account.	Y

Account/BalanceChanged	The balance of a debit or individual credit account has changed (e.g. a call has been charged, a refill has been received).  A customer's balance has changed for postpaid customers.	Y
Account/Blocked	An account has been blocked. (Accounts.blocked set to 'Y.')	
Account/CustomField / Changed	Custom information (e.g. ID card) has been added or changed for an account. (The Custom_Field_Values table has been updated for the account.)	
Account/Discount/Changed	An account's discount plan has changed.	
Account/ExpirationDate/ Changed	An account's expiration date has changed. (Accounts.expiration_date has been changed.)	
Account/FollowMe/Changed	An account's follow-me setting has changed (the Follow_Me table has been updated).	
Account/FollowMeNumber/ Inserted	A new follow-me number has been added to an account.	
Account/FollowMeNumber/ Changed	An account's follow-me number has changed.	
Account/FollowMeNumber/ Deleted	An account's follow-me number has been removed.	
Account/ID/Changed	An account's ID has changed. (Account.id has been updated.)	
Account/IPDeviceAssignment	An IP device has been assigned to an account (the UA_Links table has been updated).	
Account/New	A new account has been created.	

Account/Password/Changed	An account's password for the self-care interface has changed. (Accounts.password has been changed.)	
Account/Product/Changed	Another product was assigned to the account.  An account's product has changed. (Account.i_product has been updated.)	
Account/ProductAddon/Changed	An account's add-on product has been changed.	
Account/ProductAddon/Deleted	An add-on product has been removed from an account.	
Account/ProductAddon/Inserted	A new add-on product has been added for an account.	
Account/Service/QuotaAvailable	The account's service usage quota has been restored (quota counter has been reset or user topped up their service wallet).	
Account/ServiceAttribute/Changed	A service feature has been enabled / disabled for an account (the Service_Attribute_Values table has been updated).	Y
Account/ServiceFlags/Changed	An account's service features have changed. (Accounts.service_flags has been updated.)	Y
Account/Service/QuotaExceeded	An account has exceeded its service usage quota.	
Account/ServicePassword/Changed	An account's service password has been changed.(Accounts.h323_password has been changed.)	
Account/SIMCardAssignment	A SIM card has been assigned to, removed from or changed for an account.	

Account/Status/Changed	<p>An account's billing status has been changed or affected by the customer's billing status.</p> <p>A customer's status has changed. (Customers.bill_status has changed.)</p>	Y
Account/Status/Closed	<p>The status of the account has changed to Closed.</p> <p>The account has been removed from the web interface. (Accounts.bill_status set to 'C.')</p> <p>This event type appears when a customer is terminated.</p>	Y
Account/Status/Exported	<p>An account has been exported by Porter in dual-version PortaSwitch®. (Customers.bill_status set to E.)</p>	Y
Account/Status/Imported	<p>An account has been imported by Porter in dual-version PortaSwitch®.</p>	Y
Account/Status/Suspend	<p>An account's status has changed to suspended because of a customer's suspension.</p> <p>Caused by: Customer has unpaid invoice. (Customers.bill_status set to 'S.')</p>	Y



Account/Status/Unsuspend	An account has been unsuspended; due to removing of the customer's suspension.  A customer's account has been reinstated. (Customers.bill_status set to 'O' from 'S'.)	Y
Account/Unblocked	An account has been unblocked. (Accounts.blocked set to 'N'.)	
Account/ZeroAvailableFunds	An account's available funds have been depleted. (Accounts.balance has reached credit_limit [for credit accounts] or 0 [for debit accounts].)	Y
Custom/Jasper/CapReached	External event received from Jasper Platform.	
Customer/AvailableFunds Appear	A customer payment has been received (a customer topped up their balance or paid their invoice).	
Customer/Blocked	A customer's account has been blocked. (Customers.blocked set to 'Y'.)	
Customer/BalanceChanged	A customer's balance has been replenished (contains previous and current balance amount).	
Customer/CustomField/Changed	The data in a customer's custom field has changed (the Custom_Field_Values table has been updated for the customer).	
Customer/New	A new customer has been created.	
Customer/ServiceAttribute/Changed	A customer's service setting has changed (the Service_Attribute_Values table has been updated).	

Customer/ServiceFlags/Changed	A customer's service features have changed. (Customers.service_flags has been updated.)	
Customer/Status/Activated	A customer's record has been activated. (Customers.bill_status set to 'O' from 'D'.)	
Customer/Status/Changed	A customer's status has changed. (Customers.bill_status has been updated.)	
Customer/Status/Closed	A customer has been terminated and their status changed to closed. (Customers.bill_status set to 'C'.)	
Customer/Status/Deactivated	A customer's record has been deactivated. (Customers.bill_status set to 'D' from 'O'.)	
Customer/Status/Exported	A customer's record status has changed. (Customers.bill_status set to E.)	
Customer/Status/Imported	A customer's record status has changed. (Customers.bill_status set to O.)	
Customer/Status/Suspend	A customer's account has been suspended. (Customers.bill_status has been updated.)	
Customer/Status/Unsuspend	A customer's account has been reinstated. (Customers.bill_status set to 'O' from 'S'.)	
Customer/Unblocked	A customer's account has been unblocked. (Customers.blocked set to 'N'.)	
Customer/ZeroAvailableFunds	A customer's available funds have been depleted. (Customers.balance has reached credit_limit.)	

Subscriber/Name/Changed	The account's user name has changed.	
CustomField/Changed	A customer's custom data has changed (the Custom_Field_Values table has been updated).	
DID/New	A new DID number has been added to the DID inventory.	
DID/Status/Assigned	A DID number has been assigned to a customer.	
DID/Status/Unassigned	A DID number has been removed from a customer.	
DID/Status/Activated	A DID number has been assigned to a customer's account ID.	
DID/Status/Canceled	A DID number has been unassigned from a customer's account ID.	
DID/Status/Moved	A DID number has been moved to another installation as a result of provisioning.	
DID/Deleted	A DID number has been removed from the DID inventory.	
Invoice/Adjustment/Changed	An invoice has been adjusted (refunded) by an administrator.	
Invoice/AmountPaid/Changed	This is the amount a customer has paid. Use this event to track and provision: <ul style="list-style-type: none"> <li>- manual payment information;</li> <li>- partially paid invoice changes.</li> </ul>	
Invoice/New	A new invoice has been generated.	

Invoice/Status/Changed	<p>An invoice has changed its status from:</p> <ul style="list-style-type: none"> <li>- unpaid to paid;</li> <li>- unpaid to partially paid;</li> <li>- partially paid to paid in full.</li> </ul> <p>Note that if a customer pays their invoice in installments, the Invoice/Status/Changed event occurs twice for this invoice:</p> <ul style="list-style-type: none"> <li>- first event – changes from unpaid to partially paid;</li> <li>- second – changes from partially paid to paid in full.</li> </ul> <p>For example, a customer pays their \$110 invoice in installments (e.g. \$25+\$20+\$35+\$30). Events occur when:</p> <ul style="list-style-type: none"> <li>- the first \$25 payment is made: the invoice status changes from unpaid to partially paid;</li> </ul> <p>the last \$30 payment is made: the invoice status changes from partially paid to paid in full.</p>	
IPDeviceProfile/New	A CPE profile has been created.	
PaymentTransaction/New	A customer has made a credit card payment.	
PaymentTransaction/ResultCode/Changed	A transaction has either started or finished.	
PaymentTransaction/Status/Changed	A transaction has changed its status.	
Product/Discount/Changed	A discount plan has been changed for a product or an add-on product.	

Product/ServiceAttribute/ Changed	A service feature has been enabled / disabled for a product or an add-on product.	
ServiceAttribute/Changed	A service feature has been enabled / disabled for either a product, add-on product, customer or account.	
Subscriber/Address/Changed	The address of a user registered as an account has changed.	
Subscriber/ContactInfo/ Changed	The account's contact information has been changed.	
Subscriber/Name/Changed	The user's name registered as an account has been changed.	

## Filters

The ServiceAttributeDispatcher event handler generates provisioning events with each change in service configuration. For some services (e.g. LTE), it must only provision changes related to that service. Thus, if an account uses Internet access and an LTE service, provisioning events must be triggered only when the LTE service configuration changes.

The flexible filtering mechanism does just that. When enabled for the ServiceAttributeDispatcher event handler, it only tracks the changes of particular service attributes and triggers provisioning events. This reduces the number of queries to the database and speeds up provisioning.

### Supported filters:

#### *Ite*

This filter is applicable to LTE-related handlers: AricentHSS, Kleos::HSS, Kleos::PCRF, Huawei::HSS, Huawei::PCRF, Titan::HSS, ZXUN::HSS, ZXUN::SPR, YateHSS, Protei::PCRF.

When enabled, it tracks only the changes in the Internet access Policy and their respective attributes: Internet Access Policy name, PDP / PDN Context ID. It also tracks changes of the Static IP service feature. Changes within all other service features and attributes are ignored.

Enable this filter only if you do not need to provision other service attribute changes (e.g. for IPTV services, etc.) and have not enabled other handlers that subscribe to them.

To enable a filter, execute the following command:

```
/home/provisioning-framework/utils/evctl.pl handler filter
ServiceAttributeDispatcher lte
```

To disable a filter, execute the following command:

```
/home/provisioning-framework/utils/evctl.pl handler filter
ServiceAttributeDispatcher none
```

As a result, the filter name is displayed right after the handler name:

```
/home/provisioning-framework/utils/evctl.pl handler list
enabled
```

ID	Handler	Enabled	Description
3	ServiceAttributeDispatcher lte	Yes	Dispatch events to Accounts / Customers
122	Huawei::HSS	Yes	Huawei HSS LTE provisioning
123	Huawei::PCRF	Yes	Huawei PCRF LTE provisioning

## Configuration particularities

Consider the following points when configuring the ESPF:

- Enabling triggers on events causes additional load to the database;
- Each triggered event can be processed by more than one event handler. In this case, the event will be removed from the Event\_Queue table only after the last handler processes it.
- To subscribe an event handler to all events that it supports execute the following command:  

```
/home/provisioning-framework/utils/evctl.pl handler
sub <event_handler_name>
```
- To see the ESPF help, execute the following command:  

```
/home/provisioning-framework/utils/evctl.pl --help
```
- To ensure that your custom modifications remain on your servers during a software upgrade, list them in the Deposits page on the Configuration server web interface.

## An example of a request processing script

Below is a 7-line script that was created using Flask (a python microframework). Execute the following command in Linux to install Flask using **pip** (a package management system used to install and manage software packages written in Python):

```
sudo pip install flask
```

The script receives requests with data from a handler and writes the data into the log file:

```
from flask import Flask, request
app = Flask(__name__)
```

---

```
@app.route('/test', methods=['POST', 'GET'])
def test():
    print(request.get_data())
    return "<h1>OK</h1>"
app.run(debug=True, port=3000, host="0.0.0.0")
```

# 4. Appendices



## Appendix A. Examples of RADIUS communication

### Failed login (Authentication failure)

#### Authentication request

```
NAS-IP-Address = '193.38.81.226'
User-Name = '11380441234567'
Calling-Station-Id = '11380441234567'
Service-Type = '15'
h323-conf-id = 'BEEF0960 159C48A3 5CA4966B 7536B52C'
call-id = '94fef899-3f606c5c@192.168.192.59'
Digest-Username = '11380441234567'
Digest-Realm = '193.38.81.226'
Digest-Nonce = '1373028145:4ed5168d6120ae148bc32520fdf6bba2'
Digest-URI = 'sip:193.38.81.226'
Digest-Method = 'REGISTER'
Digest-Algorithm = 'MD5'
Digest-Response = '1984af4a0e13ceb54ae86e015e934bc5'
h323-remote-address = '192.168.192.59'
NAS-Port = '5060'
```

#### Authentication reject response

```
Reply-Message = Failed - Invalid Account number
h323-return-code = 1
h323-ivr-in = ErrorExplanation:invalid_account
```

### Failed outgoing call (Authorization failure)

#### Authorization request

```
NAS-IP-Address = '193.38.81.226'
User-Name = '380441234567'
Called-Station-Id = '91245684289'
Calling-Station-Id = '380441234567'
h323-conf-id = 'BC0050CE E4B011E2 B062000C 29E9476D'
call-id = 'f7442be6-3c18ddd@192.168.192.59'
Digest-Username = '380441234567'
Digest-Realm = '193.38.81.226'
Digest-Nonce = '51bec9bb7e2276edaf48b8572a3f72300ec0'
Digest-URI = 'sip:91245684289@193.38.81.226'
Digest-Method = 'INVITE'
Digest-Algorithm = 'MD5'
Digest-Response = '35139c9d1c67dd3192a02158d93eea50'
h323-remote-address = '192.168.192.59'
h323-session-protocol = 'sipv2'
h323-ivr-out = 'PortaBilling_AuthMethod:INVITE'
PortaOne-Service-Features = 'routing=1;version=25.0'
h323-ivr-out = 'PortaBilling_Seed:2912511541'
PortaOne-Calling-Party = 'name=380441234567'
h323-ivr-out =
'PortaSIP_Aux_Info:auth_life=134455/1/0;auth_cnt=1;CC_now=13445
5/1;pid=17785;request_type=routing'
NAS-Port = '5060'
```

## Authorization reject response

```

Reply-Message = Failed - Called number blocked
h323-return-code = 9
h323-ivr-in = ErrorExplanation:cld_blocked
h323-ivr-in = Product:EasyCall Residential
h323-ivr-in = Tariff:EasyCall Residential
h323-ivr-in = PortaBilling_NodeId:193.38.81.226
h323-ivr-in = PortaBilling_AccessCode:OUTGOING
h323-ivr-in = PortaBilling_Seed:1599467173
h323-billing-model = 0
h323-currency = USD
h323-preferred-lang = en

```

## Successful prepaid card IVR session

### Authentication request

```

NAS-IP-Address      = '164.9.9.100'
NAS-Port-Type       = 'Async'
User-Name           = '10086610975'
Calling-Station-Id  = '6045550193'
Service-Type        = 'Login-User'
h323-conf-id        = '39AE126B CD4D11DB 958E0014 1C3F6886'
Password            = ''
h323-ivr-out        = 'PortaBilling_AccessCode:6045551600'
h323-ivr-out        = 'transactionID:526267'
NAS-Port            = '0'
NAS-Port-Id         = 'ISDN 3/0:D:13'

```

### Authentication response

```

h323-credit-amount  = 10.00
h323-billing-model   = 1
h323-ivr-in          = Tariff:PrepaidCard
h323-ivr-in          = PortaBilling_AccountBalance:10.00000
h323-ivr-in          = PortaBilling_ProductBreakage:0.00000
h323-ivr-in          = available-funds:10.00
h323-return-code     = 0
h323-currency        = CAD
h323-preferred-lang  = en

```

### Authorization request

```

NAS-IP-Address      = '164.9.9.100'
User-Name           = '10086610975'
Called-Station-Id   = '01182623634515'
Calling-Station-Id  = '6045550193'
Service-Type        = 'Login-User'
h323-conf-id        = '39AE126B CD4D11DB 958E0014 1C3F6886'
Password            = ''
h323-ivr-out        = 'PortaBilling_Seed:18879071672'
h323-ivr-out        = 'PortaBilling_Original_CLD:6045551600'
h323-ivr-out        = 'transactionID:526273'

```

### Authorization response

```

h323-billing-model   = 1
h323-ivr-in          = Tariff:PrepaidCard
h323-ivr-in          = PortaBilling_CompleteNumber:82623634515
h323-ivr-in          = PortaBilling_Auth_CLD:82623634515
h323-ivr-in          = PortaBilling_Auth_Reseller_CLD:82623634515
h323-ivr-in          = DURATION:30000
h323-return-code     = 0

```

```

h323-currency           = CAD
h323-credit-time        = 30000
h323-preferred-lang     = en

```

### Accounting request (outgoing call leg)

```

NAS-IP-Address          = '164.9.9.100'
User-Name               = '10086610975'
Called-Station-Id       = '82623634515'
Calling-Station-Id      = '6045550193'
Acct-Status-Type        = 'Stop'
Service-Type            = 'Login-User'
h323-call-origin        = 'originate'
h323-call-type          = 'VoIP'
h323-setup-time         = '00:16:18.192 PST Fri Mar 9 2007'
h323-connect-time       = '00:16:21.164 PST Fri Mar 9 2007'
h323-disconnect-time    = '00:17:31.893 PST Fri Mar 9 2007'
h323-disconnect-cause   = '10'
h323-conf-id            = '39AE126B CD4D11DB 958E0014 1C3F6886'
call-id                 = '4A65E46C-CD4D11DB-886EDEBF-'
7AF0CBAB@164.9.9.100'
h323-incoming-conf-id   = '39AE126B CD4D11DB 958E0014 1C3F6886'
h323-ivr-out            = 'PortaBilling_Original_CLD:6045551600'
h323-ivr-out            = 'PortaBilling_Seed:18879071672'
h323-remote-address     = '164.9.9.101'
remote-media-address    = '164.9.9.101'

```

### Accounting request (incoming call leg)

```

NAS-IP-Address          = '164.9.9.100'
NAS-Port-Type           = 'Async'
User-Name               = '10086610975'
Called-Station-Id       = '6045551600'
Calling-Station-Id      = '6045550193'
Acct-Status-Type        = 'Stop'
Service-Type            = 'Login-User'
h323-call-origin        = 'answer'
h323-call-type          = 'Telephony'
h323-setup-time         = '00:15:50.148 PST Fri Mar 9 2007'
h323-connect-time       = '00:15:50.156 PST Fri Mar 9 2007'
h323-disconnect-time    = '00:17:31.889 PST Fri Mar 9 2007'
h323-disconnect-cause   = '10'
h323-conf-id            = '39AE126B CD4D11DB 958E0014 1C3F6886'
h323-incoming-conf-id   = '39AE126B CD4D11DB 958E0014 1C3F6886'
h323-ivr-out            = 'PortaBilling_Session:unlock'
h323-ivr-out            = 'Tariff:Unknown'
NAS-Port                = '0'
NAS-Port-Id             = 'ISDN 3/0:D:13'

```

## SIP routing request

### Authorization request

```

NAS-IP-Address = '193.28.87.217'
User-Name      = '193.38.81.226'
Called-Station-Id = '12125468754'
Calling-Station-Id = '380441234567'
h323-conf-id = 'FD0158BA E2F911E2 B062000C 29E9476D'
call-id = '482bb2cb-aa0eada5@192.168.192.59~1o'
Password = 'cisco'
h323-remote-address = '193.38.81.226'
h323-session-protocol = 'sipv2'
h323-ivr-out = 'PortaBilling_AuthMethod:INVITE'
PortaOne-Service-Features = 'routing=1;version=25.0'
h323-ivr-out = 'PortaBilling_Seed:2567011031'
PortaOne-Calling-Party = 'id=380441234567;name=11380441234567'

```

```
h323-ivr-out = 'PortaBilling_Ignore_Password:YES'
h323-ivr-out =
'PortaSIP_Aux_Info:auth_life=164498/1/0;auth_cnt=1;CC_now=16449
8/1;pid=19218;request_type=routing'
NAS-Port = '5060'
```

### Authorization response

```
h323-return-code = 13
h323-ivr-in = PortaBilling_CompleteNumber:12125468754
h323-ivr-in = Product:PortaOne Internal
h323-ivr-in = Tariff:Internal
h323-ivr-in = PortaBilling_NodeId:193.38.81.217
h323-ivr-in = PortaBilling_AccessCode:OUTGOING
h323-ivr-in = PortaBilling_Seed:2567011031
h323-ivr-in = PortaBilling_RatePattern:12125468754
h323-ivr-in = PortaBilling_Authorize:base=session-time;avail=-
1;expires=1373028131
h323-billing-model = 0
PortaOne-Service-Features = moh=1;rec=2;rtp-level=3
PortaOne-Calling-Party =
name=11380441234567;id=380441234567;display-id=380441234567
h323-currency = USD
h323-preferred-lang = en
Framed-Route = r;;g-hunt=seq
Framed-Route = P_;;g-hunt=skip;expires=300;patience=20
Framed-Route =
p__=65.174.119.133;auth=AA14CF3595B86BF7B1C7B0458626C288DD5B419
1C0B759A3
Framed-Route = p__=69.25.184.13;rtp-level=3
```

## Callback session

### First call leg authorization

```
NAS-IP-Address      = "142.179.71.220"
User-Name           = "00099900113"
Called-Station-Id   = "00099900113"
Calling-Station-Id  = "00099900113"
h323-conf-id        = "2F01F3F6 B89192B8 711B0000 1C7A0F2A "
Password            = "secret"
h323-remote-address = "193.28.87.193"
h323-ivr-out         = "PortaBilling_OriginalCLD:555333"
h323-ivr-out         = "PortaBilling_AuthMethod:INVITE"
h323-ivr-out         = "PortaBilling_Seed:4094388346"
Message-Authenticator = '84C47B0BDC12887A60D2C5167C740A'
```

### Second call leg authorization

```
NAS-IP-Address      = "142.179.71.220"
User-Name           = "00099900113"
Called-Station-Id   = "00099900111"
Calling-Station-Id  = "00099900222"
h323-conf-id        = "2F01F3F6 B89192B8 711B0000 1C7A0F2A"
Password            = "secret"
h323-ivr-out         = "PortaBilling_OriginalCLD:OUT"
h323-ivr-out         = "PortaBilling_AuthMethod:INVITE"
h323-ivr-out         =
PortaBilling_Notify:another_cld=00099900113;another_orig_cld=55
5333;in_progress=13"
h323-ivr-out         = "PortaBilling_Seed:4094388346"
```

Message-Authenticator = '12C47B0B16412247A60D2C5167C718C'

### Second call leg 1, accounting for failed call attempt

```
NAS-IP-Address           = 142.179.71.220
h323-call-origin         = "originate"
h323-call-type           = "VoIP"
h323-setup-time          = "11:34:20.677 GMT Fri Feb 23 2007"
User-Name                 = "00099900113"
Calling-Station-Id       = "00099900113"
Called-Station-Id        = "000999002222"
h323-incoming-conf-id    = "2F01F3F6 B89192B8 711B0000 1C7A0F2A"
h323-conf-id             = "11111111 B89192B8 711B0000 1C7A0F2A"
Acct-Session-Id          = "B89192B8-66BC43D89"
h323-disconnect-time     = "11:34:30.677 GMT Fri Feb 23 2007"
h323-connect-time        = "11:34:30.677 GMT Fri Feb 23 2007"
h323-ivr-out             = "PortaBilling_Seed:4094388346"
h323-remote-address      = "10.28.87.193"
h323-ivr-out             = "PortaBilling_OriginalCLD:OUT"
Acct-Session-Time        = 0
h323-disconnect-cause    = "1C"
Acct-Status-Type         = Stop
```

### Second call leg, accounting for connected call

```
NAS-IP-Address           = 142.179.71.220
h323-call-origin         = "originate"
h323-call-type           = "VoIP"
h323-setup-time          = "11:34:35.677 GMT Fri Feb 23 2007"
User-Name                 = "00099900113"
Calling-Station-Id       = "00099900113"
Called-Station-Id        = "000999002222"
h323-incoming-conf-id    = "2F01F3F6 B89192B8 711B0000 1C7A0F2A"
h323-conf-id             = "11111111 B89192B8 711B0000 1C7A0F2A"
Acct-Session-Id          = "B89192B8-66BC43D89"
h323-disconnect-time     = "11:34:30.677 GMT Fri Feb 23 2007"
h323-connect-time        = "11:36:32.677 GMT Fri Feb 23 2007"
h323-ivr-out             = "PortaBilling_Seed:4094388346"
h323-remote-address      = "10.28.87.200"
h323-ivr-out             = "PortaBilling_OriginalCLD:OUT"
Acct-Session-Time        = 122
h323-disconnect-cause    = "0"
Acct-Status-Type         = Stop
```

### First call leg accounting

```
NAS-IP-Address           = 142.179.71.220
h323-call-origin         = "originate"
h323-call-type           = "VoIP"
h323-setup-time          = "11:33:38.677 GMT Fri Feb 23 2007"
User-Name                 = "00099900113"
Calling-Station-Id       = "00099900113"
Called-Station-Id        = "00099900113"
h323-ivr-out             = PortaBilling_CallbackHistory:START
h323-ivr-out             = PortaBilling_CallbackHistory:ATTEMPT
h323-ivr-out             = PortaBilling_CallbackHistory:ATTEMPT
h323-ivr-out             = PortaBilling_CallbackHistory:OK
h323-conf-id             = "2F01F3F6 B89192B8 711B0000 1C7A0F2A"
Acct-Session-Id          = "B89192B8-66BC43D89"
h323-disconnect-time     = "11:34:18.677 GMT Fri Feb 23 2007"
h323-connect-time        = "11:35:18.677 GMT Fri Feb 23 2007"
h323-ivr-out             = "PortaBilling_Seed:4094388346"
h323-remote-address      = "10.28.87.200"
h323-ivr-out             = "PortaBilling_OriginalCLD:555333"
Acct-Session-Time        = 60
h323-disconnect-cause    = "0"
Acct-Status-Type         = Stop
Acct-Delay-Time          = 0
```

## Voucher recharge session

### Authentication (main account login) request

```
NAS-IP-Address      = '127.0.0.102'
User-Name           = '02001'
Calling-Station-Id  = '02001'
h323-conf-id        = '01010101 00000000 00000000 00000012'
PortaOne-Service-Type = 'Quantity'
h323-ivr-out         = 'PortaBilling_Original_CLD:01inbound'
h323-ivr-out = 'PortaBilling_Ignore_Password:YES'
Message-Authenticator = '52C47B0B16412147A60D2C5167C718D'
```

### Authentication response

```
h323-credit-amount  = 10.00
h323-billing-model   = 1
h323-ivr-in          = PortaBilling_AccessCode:01inbound
h323-ivr-in          = Tariff:T_SubService_1
h323-ivr-in          = PortaBilling_AccountBalance:10.00000
h323-ivr-in          = PortaBilling_ProductBreakage:0.00000
h323-ivr-in          = available-funds:10.00
h323-return-code     = 0
h323-currency        = USD
h323-preferred-lang  = en
Message-Authenticator = '1CD6D5B0BDC33987A60D5C6167B22517'
```

### Voucher recharge request

```
NAS-IP-Address      = '127.0.0.102'
User-Name           = '02001'
Calling-Station-Id  = '02001'
h323-conf-id        = '01010101 00000000 00000000 00000012'
h323-conf-id = 'PortaBilling_Session:ignore'
h323-ivr-out =
'PortaBilling_Mod_ivrutil:action=rechargeVoucher;id=111222336'
h323-ivr-out         = 'PortaBilling_Original_CLD:01inbound'
h323-ivr-out = 'PortaBilling_Ignore_Password:YES'
```

### Voucher recharge response

```
h323-ivr-in          = PortaBilling_AccessCode:01inbound
h323-ivr-in =
'PortaBilling_Mod_ivrutil:action=rechargeVoucher;amount=50.00000'
h323-ivr-in          = PortaBilling_RechargedAmount:50.00000
h323-ivr-in          = Tariff:T_SubService_1
h323-ivr-in          = available-funds:60.00
h323-return-code     = 0
h323-credit-amount   = 60.00
h323-currency        = USD
h323-preferred-lang  = en
```

## Music download service requests

### Authorization request

```
NAS-IP-Address      = '192.168.100.50'
User-Name           = '0101'
Password            = 'secret'
Called-Station-Id   = 'JAZZ'
h323-conf-id        = 'A5D3B74C D26411DC 98D5001D 090C82FA'
PortaOne-Service-Type = Quantity
```

### Authorization response (for credit account without credit limit)

```

h323-billing-model    = 0
h323-ivr-in           = Tariff:Music Account
h323-ivr-in           = PortaBilling_CompleteNumber:JAZZ
h323-ivr-in           = PortaBilling_Auth_CLD:JAZZ
h323-return-code      = 13
h323-currency         = USD
h323-preferred-lang   = en
Message-Authenticator = '1D86D5B0BDC33987A60D5C6167B13D91'

```

### Authorization response (for debit account)

```

h323-billing-model    = 1
h323-ivr-in           = Tariff:Music Account
h323-ivr-in           = DURATION:10
h323-ivr-in           = PortaBilling_CompleteNumber:JAZZ
h323-ivr-in           = PortaBilling_Auth_CLD:JAZZ
h323-return-code      = 0
h323-currency         = USD
h323-credit-time      = 10
h323-preferred-lang   = en

```

### Accounting request

```

NAS-IP-Address        = '192.168.100.50'
User-Name              = '0101'
Called-Station-Id     = 'JAZZ'
Acct-Status-Type      = 'Stop'
h323-call-origin      = 'originate'
h323-call-type        = 'Telephony'
h323-setup-time       = '07:18:35.965 GMT Mon Feb 6 2008'
h323-connect-time     = '07:18:43.965 GMT Mon Feb 6 2008'
h323-disconnect-time  = '08:11:38.965 GMT Mon Feb 6 2008'
h323-disconnect-cause = '0'
h323-conf-id          = 'A5D3B74C D26411DC 98D5001D 090C82FA'
Acct-Session-Id       = 'da6b76'
Acct-Session-Time     = '3175'
PortaOne-Service-Type = Quantity
h323-ivr-out          = 'Used-Service-Unit:1'

```

## Internet session

### Authorization request

```

NAS-IP-Address        = '41.213.1.32'
NAS-Port-Type         = 'Ethernet'
User-Name              = 'GP4899@ibhighspeed.co.za'
Called-Station-Id     = 'pppoe-service1'
Calling-Station-Id    = 'E4:8D:8C:EE:1A:26'
Service-Type          = 'Framed-User'
Mikrotik-Realm        = 'ibhighspeed.co.za'
Unknown-Attribute     = 'unknown vendor: attr
{311, 10, 0, 17, "ibhighspeed.co.za"}'
Framed-Protocol       = 'PPP'
Event-Timestamp       = '1514299872'
NAS-Identifier        = 'WDSL_PPPE2_MDC'
Password              = 'password1'
NAS-Port              = '15760440'
NAS-Port-Id           = 'vlan140'

```

**Authorization response**

```

Service-Type                = 'Framed-User'
Session-Timeout              = '0'
Ascend-Data-Rate             = '20000000'
Idle-Timeout                 = '86400'
Framed-IP-Address            = '41.213.82.27'
Ascend-Xmit-Rate             = '20000000'
Framed-IP-Netmask            = '255.255.255.255'
Framed-MTU                   = '1500'

```

**Start Accounting request**

```

NAS-IP-Address               = '41.213.1.32'
NAS-Port-Type                = 'Ethernet'
User-Name                    = 'GP4899@ibhighspeed.co.za'
Called-Station-Id            = 'pppoe-service1'
Calling-Station-Id           = 'E4:8D:8C:EE:1A:26'
Acct-Status-Type             = 'Start'
Service-Type                 = 'Framed-User'
Acct-Session-Id              = '81707b87'
Acct-Delay-Time              = '0'
Mikrotik-Realm               = 'ibhighspeed.co.za'
Unknown-Attribute            = 'unknown vendor: attr
{311, 10, 0, 17, "ibhighspeed.co.za"}'
Acct-Authentic               = 'RADIUS'
NAS-Port-Id                  = 'vlan140'
Framed-Protocol              = 'PPP'
NAS-Identifier                = 'WDSL_PPPoE2_MDC'
Event-Timestamp              = '1514299872'
Framed-IP-Address            = '41.213.82.27'
NAS-Port                     = '15760440'

```

**Keep-Alive Accounting request**

```

NAS-IP-Address               = '41.213.1.32'
NAS-Port-Type                = 'Ethernet'
User-Name                    = 'GP4899@ibhighspeed.co.za'
Called-Station-Id            = 'pppoe-service1'
Calling-Station-Id           = 'E4:8D:8C:EE:1A:26'
Acct-Status-Type             = 'Alive'
Service-Type                 = 'Framed-User'
Acct-Session-Id              = '81707b87'
Acct-Input-Octets            = '1013118'
Acct-Output-Octets           = '599107'
Acct-Input-Packets           = '3494'
Acct-Output-Packets          = '3082'
Acct-Session-Time            = '1800'
Acct-Delay-Time              = '0'
Mikrotik-Realm               = 'ibhighspeed.co.za'
Unknown-Attribute            = 'unknown vendor: attr
{311, 10, 0, 17, "ibhighspeed.co.za"}'
Idle-Timeout                 = '86400'
NAS-Identifier                = 'WDSL_PPPoE2_MDC'
Framed-IP-Address            = '41.213.82.27'
Session-Timeout              = '0'
Unknown-Attribute            = 'unknown code: attr {0,
197, 0, 4, "\0011-\000"}'
Acct-Output-Gigawords        = '0'
Acct-Authentic               = 'RADIUS'
NAS-Port-Id                  = 'vlan140'

```



```

Framed-Protocol          = 'PPP'
Event-Timestamp          = '1514301672'
Acct-Input-Gigawords     = '0'
Ascend-Xmit-Rate         = '20000000'
NAS-Port                 = '15760440'

```

#### Stop Accounting request

```

NAS-IP-Address           = '41.213.1.32'
NAS-Port-Type            = 'Ethernet'
User-Name                = 'GP4899@ibhighspeed.co.za'
Called-Station-Id        = 'pppoe-service1'
Calling-Station-Id       = 'E4:8D:8C:EE:1A:26'
Acct-Status-Type         = 'Stop'
Service-Type             = 'Framed-User'
Acct-Session-Id          = '81707b87'
Acct-Input-Octets        = '2668094'
Acct-Output-Octets       = '65615636'
Acct-Input-Packets       = '20697'
Acct-Output-Packets      = '52139'
Acct-Session-Time        = '3725'
Acct-Delay-Time          = '0'
Mikrotik-Realm           = 'ibhighspeed.co.za'
Unknown-Attribute        = 'unknown vendor: attr
{311, 10, 0, 17, "ibhighspeed.co.za"}'
Idle-Timeout             = '86400'
Acct-Terminate-Cause     = 'Lost-Carrier'
NAS-Identifier            = 'WDSL_PPpOE2_MDC'
Framed-IP-Address        = '41.213.82.27'
Session-Timeout          = '0'
Unknown-Attribute        = 'unknown code: attr {0,
197, 0, 4, "\0011-\000"}'
Acct-Output-Gigawords    = '0'
Acct-Authentic           = 'RADIUS'
NAS-Port-Id              = 'vlan140'
Framed-Protocol          = 'PPP'
Event-Timestamp          = '1514303597'
Acct-Input-Gigawords     = '0'
Ascend-Xmit-Rate         = '20000000'
NAS-Port                 = '15760440'

```

## xDR import post-processing

### Accounting request, originate leg

```

NAS-IP-Address = 127.0.0.101
NAS-Port-Id = IMPORT 01
User-Name = 01004
Called-Station-Id = 01020
Calling-Station-Id = 01004
Acct-Status-Type = Stop
PortaOne-Service-Type = Voice
h323-call-origin = originate
h323-call-type = Telephony
PortaOne-Service-Features = charge-report=1
h323-setup-time = 10:44:24.000 CEST Mon Mar 30 2009
h323-connect-time = 10:44:34.000 CEST Mon Mar 30 2009
h323-disconnect-time = 10:48:24.000 CEST Mon Mar 30 2009
h323-conf-id = 01010101 00000000 00000000 16000023
Acct-Session-Time = 230
Acct-Delay-Time = 86400
h323-ivr-out = PortaBilling_RatePattern:01020

```

```
h323-ivr-out = PortaBilling_Session:unlock
```

### Accounting request, charge report:

```
h323-ivr-in = PortaBilling_Session:id=01010101 00000000
00000000 16000023;i_service_type=3;i_env=3;valid=1
PortaOne-Charge = type=account;storage-type=Account;charged-
amount=0.00005;charged-quantity=10;i_account=500055
PortaOne-Charge = type=reseller;storage-type=Customer;charged-
amount=0.00010;charged-quantity=10;i_customer=20
PortaOne-Charge = type=vendor;storage-type=Vendor;charged-
amount=0.00005;charged-quantity=11;i_vendor=57
```

## Appendix B. Examples of SIP communication

## Messaging via SIP SIMPLE

In this example a message is sent between accounts of the same IP Centrex environment using classic (non cluster) PortaSwitch® configuration.

## Part 1. Sending message

SIP Log Viewer

Close Send logs to RT Download Logout

Search query SIP Log 5 SIP Log 7

Porta SIP mode: 193.28.87.73  
 Call-ID: bcd95a2c977262271a7b0cccd52792@0.0.0.0.0.0.0  
 H323 Conf-ID: not found

Hide panel

Log level: 0 - simple

Date / Time	SIP: 193.21.214.241:87121	SIP: 193.28.87.73:8040
2016-04-13 02:21:27.905	2016-04-13 02:21:27.905	2016-04-13 02:21:27.905
2016-04-13 02:21:27.988	2016-04-13 02:21:27.988	2016-04-13 02:21:27.988
2016-04-13 02:21:27.988	2016-04-13 02:21:27.988	2016-04-13 02:21:27.988
2016-04-13 02:21:27.989	2016-04-13 02:21:27.989	2016-04-13 02:21:27.989
2016-04-13 02:21:28.000	2016-04-13 02:21:28.000	2016-04-13 02:21:28.000
2016-04-13 02:21:28.000	2016-04-13 02:21:28.000	2016-04-13 02:21:28.000

*A sender's user agent sends a message to the PortaSIP® node*

```

2016-04-
13T09:21:27.905890Z|portaproxy[10740]|bd595ab29772f82271a7ccbcecc6f2792@
0:0:0:0:0:0:0:0|IS|622| RECEIVED message from UDP:91.212.34.242:57217
at UDP:193.28.87.73:5060:
    MESSAGE sip:7008004@193.28.87.73 SIP/2.0
    Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
ce7da5393ec80f60be280c053a117f13;received=91.212.34.242
    Max-Forwards: 70
    Contact: "7008001"
<sip:7008001@192.168.31.60:22430;transport=udp;registering_acc=193_28_8
7_73>
    To: <sip:7008004@193.28.87.73>
    From: "7008001" <sip:7008001@193.28.87.73>;tag=f2e03f39
    Call-ID: bd595ab29772f82271a7ccbcecc6f2792@0:0:0:0:0:0:0:0
    CSeq: 31451132 MESSAGE
    Content-Type: text/plain
    User-Agent: Jitsi2.8.5426Windows 7
    Content-Length: 19

    Hello! How are you?

```

### *The PortaSIP® node transfers the message to the IMGate node*

```

2016-04-
13T09:21:27.906677Z|portaproxy[10740]|bd595ab29772f82271a7ccb66f2792@
0:0:0:0:0:0:0:0|IS|808| SENDING message to UDP:193.28.87.233:5960 from
UDP:193.28.87.73:5060:
  MESSAGE sip:7008004@193.28.87.233:5960 SIP/2.0
  Via: SIP/2.0/UDP 193.28.87.73:5060;branch=z9hG4bK-524287-1---
5d97e87d8fb47b33;rport
  Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
ce7da5393ec80f60be280c053a117f13;received=91.212.34.242;rport=57217
  Max-Forwards: 69
  Contact:
"7008001"<sip:7008001@91.212.34.242:57217;transport=udp;registering_acc
=193_28_87_73>
  To: <sip:7008004@193.28.87.73>
  From: "7008001" <sip:7008001@193.28.87.73>;tag=f2e03f39
  Call-ID: bd595ab29772f82271a7ccb66f2792@0:0:0:0:0:0:0:0
  CSeq: 31451132 MESSAGE
  Content-Type: text/plain
  User-Agent: Jitsi2.8.5426Windows 7
  PortaSIP-Notify: NAT=91.212.34.242
  PortaOne-Requested-URI: sip:7008004@193.28.87.73
  Content-Length: 19

Hello! How are you?

```

### *IMGate responds to the PortaSIP® node that user's authentication is required*

```

2016-04-
13T09:21:27.956696Z|portaproxy[10740]|bd595ab29772f82271a7ccb66f2792@
0:0:0:0:0:0:0:0|IS|662| RECEIVED message from UDP:193.28.87.233:5960 at
UDP:193.28.87.73:5060:
  SIP/2.0 407 Proxy Authentication Required
  Via: SIP/2.0/UDP 193.28.87.73:5060;branch=z9hG4bK-524287-1---
5d97e87d8fb47b33;rport=5060
  Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
ce7da5393ec80f60be280c053a117f13;received=91.212.34.242;rport=57217
  Proxy-Authenticate: Digest
nonce="1460539287:cd0ab87b3060a8856d1558cf6233707d",algorithm=MD5,realm
="sip-41.TechWritting"
  To: <sip:7008004@193.28.87.73>;tag=3f3aff1b
  From: "7008001" <sip:7008001@193.28.87.73>;tag=f2e03f39
  Call-ID: bd595ab29772f82271a7ccb66f2792@0:0:0:0:0:0:0:0
  CSeq: 31451132 MESSAGE
  Content-Length: 0

```

### *The PortaSIP® node sends authentication request to the sender's UA*

```

2016-04-
13T09:21:27.956965Z|portaproxy[10740]|bd595ab29772f82271a7ccb66f2792@
0:0:0:0:0:0:0:0|IS|572| SENDING message to UDP:91.212.34.242:57217 from
UDP:193.28.87.73:5060:
  SIP/2.0 407 Proxy Authentication Required
  Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
ce7da5393ec80f60be280c053a117f13;received=91.212.34.242;rport=57217
  Proxy-Authenticate: Digest
nonce="1460539287:cd0ab87b3060a8856d1558cf6233707d",algorithm=MD5,realm
="sip-41.TechWritting"
  To: <sip:7008004@193.28.87.73>;tag=3f3aff1b
  From: "7008001" <sip:7008001@193.28.87.73>;tag=f2e03f39
  Call-ID: bd595ab29772f82271a7ccb66f2792@0:0:0:0:0:0:0:0
  CSeq: 31451132 MESSAGE
  Content-Length: 0

```

*The sender's UA sends proxy authorization information to the PortaSIP® node*

```

2016-04-
13T09:21:27.989028Z|portaproxy[10740]|bd595ab29772f82271a7ccbcecc6f2792@
0:0:0:0:0:0:0:0|IS|839| RECEIVED message from UDP:91.212.34.242:57217
at UDP:193.28.87.73:5060:
  MESSAGE sip:7008004@193.28.87.73 SIP/2.0
  Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
67b34549a6d8b28b42f4488789779e2f;received=91.212.34.242
  Max-Forwards: 70
  Contact: "7008001"
<sip:7008001@192.168.31.60:22430;transport=udp;registering_acc=193_28_8
7_73>
  To: <sip:7008004@193.28.87.73>
  From: "7008001" <sip:7008001@193.28.87.73>;tag=f2e03f39
  Call-ID: bd595ab29772f82271a7ccbcecc6f2792@0:0:0:0:0:0:0:0
  CSeq: 31451133 MESSAGE
  Content-Type: text/plain
  Proxy-Authorization: Digest username="7008001",realm="sip-
41.TechWritting",nonce="1460539287:cd0ab87b3060a8856d1558cf6233707d",ur
i="sip:7008004@193.28.87.73",response="a9166ff5c75ae4fa9f6813fe581de2c4
",algorithm=MD5
  User-Agent: Jitsi2.8.5426Windows 7
  Content-Length: 19

Hello! How are you?

```

*The PortaSIP® node transfers the proxy authorization information message to IMGate*

```

2016-04-
13T09:21:27.989781Z|portaproxy[10740]|bd595ab29772f82271a7ccbcecc6f2792@
0:0:0:0:0:0:0:0|IS|1025| SENDING message to UDP:193.28.87.233:5960 from
UDP:193.28.87.73:5060:
  MESSAGE sip:7008004@193.28.87.233:5960 SIP/2.0
  Via: SIP/2.0/UDP 193.28.87.73:5060;branch=z9hG4bK-524287-1---
7e01222648b0c370;rport
  Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
67b34549a6d8b28b42f4488789779e2f;received=91.212.34.242;rport=57217
  Max-Forwards: 69
  Contact:
"7008001"<sip:7008001@91.212.34.242:57217;transport=udp;registering_acc
=193_28_87_73>
  To: <sip:7008004@193.28.87.73>
  From: "7008001" <sip:7008001@193.28.87.73>;tag=f2e03f39
  Call-ID: bd595ab29772f82271a7ccbcecc6f2792@0:0:0:0:0:0:0:0
  CSeq: 31451133 MESSAGE
  Content-Type: text/plain
  Proxy-Authorization: Digest username="7008001",realm="sip-
41.TechWritting",nonce="1460539287:cd0ab87b3060a8856d1558cf6233707d",ur
i="sip:7008004@193.28.87.73",response="a9166ff5c75ae4fa9f6813fe581de2c4
",algorithm=MD5
  User-Agent: Jitsi2.8.5426Windows 7
  PortaSIP-Notify: NAT=91.212.34.242
  PortaOne-Requested-URI: sip:7008004@193.28.87.73
  Content-Length: 19

Hello! How are you?

```

*IMGate sends the authorization request to the PortaBilling® and if the PortaBilling® authorizes the sender, informs the PortaSIP® node about this and that and the message is accepted*

```

2016-04-
13T09:21:28.090083Z|portaproxy[10740]|bd595ab29772f82271a7ccbcecc6f2792@

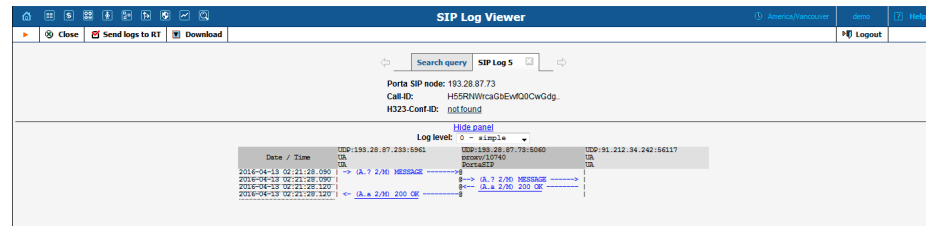
```

```
0:0:0:0:0:0:0:0|IS|518| RECEIVED message from UDP:193.28.87.233:5960 at
UDP:193.28.87.73:5060:
  SIP/2.0 202 Accepted
  Via: SIP/2.0/UDP 193.28.87.73:5060;branch=z9hG4bK-524287-1---
7e01222648b0c370;rport=5060
  Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
67b34549a6d8b28b42f4488789779e2f;received=91.212.34.242;rport=57217
  To: <sip:7008004@193.28.87.73>;tag=95ab980f
  From: "7008001"<sip:7008001@193.28.87.73>;tag=f2e03f39
  Call-ID: bd595ab29772f82271a7ccb6cc6f2792@0:0:0:0:0:0:0:0
  CSeq: 31451133 MESSAGE
  Content-Length: 0
```

*The PortaSIP® node informs the sender's UA that that the authorization is successful the message is accepted*

```
2016-04-
13T09:21:28.090353Z|portaproxy[10740]|bd595ab29772f82271a7ccb6cc6f2792@
0:0:0:0:0:0:0:0|IS|428| SENDING message to UDP:91.212.34.242:57217 from
UDP:193.28.87.73:5060:
  SIP/2.0 202 Accepted
  Via: SIP/2.0/UDP 192.168.31.60:22430;branch=z9hG4bK-333636-
67b34549a6d8b28b42f4488789779e2f;received=91.212.34.242;rport=57217
  To: <sip:7008004@193.28.87.73>;tag=95ab980f
  From: "7008001"<sip:7008001@193.28.87.73>;tag=f2e03f39
  Call-ID: bd595ab29772f82271a7ccb6cc6f2792@0:0:0:0:0:0:0:0
  CSeq: 31451133 MESSAGE
  Content-Length: 0
```

## Part 2. Delivering message



*IMGate transfers the message to a recipient's SIP server*

```
2016-04-
13T09:21:28.090215Z|portaproxy[10740]|H55RNWrcAGbEvvfQ0CwGdg..|IS|500|
RECEIVED message from UDP:193.28.87.233:5961 at UDP:193.28.87.73:5060:
  MESSAGE sip:7008004@193.28.87.73:5060 SIP/2.0
  Via: SIP/2.0/UDP 193.28.87.233:5961;branch=z9hG4bK-524287-1---
834c3c3758bb016b;rport=5961
  Max-Forwards: 70
  Route: <sip:193.28.87.73:5060;lr>
  To: <sip:7008004@193.28.87.73:5060;lr>
  From: <sip:7008001@193.28.87.233:5961>;tag=5b0b2f73
  Call-ID: H55RNWrcAGbEvvfQ0CwGdg..
  CSeq: 2 MESSAGE
  Allow: MESSAGE, NOTIFY
  Content-Type: text/plain
  Content-Length: 19

Hello! How are you?
```

*The recipient's SIP server transfers the message to the recipient's UA*

```
2016-04-
13T09:21:28.090539Z|portaproxy[10740]|H55RNWrcAGbEvvfQ0CwGdg..|IS|569|
SENDING message to UDP:91.212.34.242:56117 from UDP:193.28.87.73:5060:
```

```

MESSAGE sip:7008004@91.212.34.242:56117;ob;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 193.28.87.73:5060;branch=z9hG4bK-524287-1---
45fb5d774b48af55;rport
Via: SIP/2.0/UDP 193.28.87.233:5961;branch=z9hG4bK-524287-1---
834c3c3758bb016b;rport=5961
Max-Forwards: 69
To: <sip:7008004@193.28.87.73:5060;lr>
From: <sip:7008001@193.28.87.233:5961>;tag=5b0b2f73
Call-ID: H55RNWrcaGbEvvfQ0CwGdg..
CSeq: 2 MESSAGE
Allow: MESSAGE, NOTIFY
Content-Type: text/plain
Content-Length: 19

Hello! How are you?

```

*The recipient's UA responds to the recipient's SIP server that the message is accepted*

```

2016-04-
13T09:21:28.120507Z|portaproxy[10740]|H55RNWrcaGbEvvfQ0CwGdg..|IS|486|
RECEIVED message from UDP:91.212.34.242:56117 at UDP:193.28.87.73:5060:
SIP/2.0 200 OK
Via: SIP/2.0/UDP
193.28.87.73:5060;rport=5060;received=193.28.87.73;branch=z9hG4bK-
524287-1---45fb5d774b48af55
Via: SIP/2.0/UDP 193.28.87.233:5961;rport=5961;branch=z9hG4bK-
524287-1---834c3c3758bb016b
To: <sip:7008004@193.28.87.73>;tag=z9hG4bK-524287-1---
45fb5d774b48af55
From: <sip:7008001@193.28.87.233>;tag=5b0b2f73
Call-ID: H55RNWrcaGbEvvfQ0CwGdg..
CSeq: 2 MESSAGE
Content-Length: 0

```

*The recipient's SIP server informs IMGate that the message is accepted*

```

2016-04-
13T09:21:28.120760Z|portaproxy[10740]|H55RNWrcaGbEvvfQ0CwGdg..|IS|372|
SENDING message to UDP:193.28.87.233:5961 from UDP:193.28.87.73:5060:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 193.28.87.233:5961;rport=5961;branch=z9hG4bK-
524287-1---834c3c3758bb016b
To: <sip:7008004@193.28.87.73>;tag=z9hG4bK-524287-1---
45fb5d774b48af55
From: <sip:7008001@193.28.87.233>;tag=5b0b2f73
Call-ID: H55RNWrcaGbEvvfQ0CwGdg..
CSeq: 2 MESSAGE
Content-Length: 0

```

## Appendix C. Description of xDR record fields

Field	Type	Description
<i>id</i>	int	The unique ID of the xDR
<i>i_env</i>	int	The unique ID of the virtual billing environment
<i>h323_conf_id</i> ( <i>session_id</i> )	string	The unique session ID

<i>call_id</i> ( <i>service_session_id</i> )	string	The session ID specific for the xDR's service type
<i>CLI</i> ( <i>cli</i> )	string	The caller number (also called CLI or ANI)
<i>CLD</i> ( <i>cld</i> )	string	The destination number (also called CLD, Called-Station-Id or DNIS)
<i>setup_time</i>	int	Also known as <b>PDD</b> . The time (in ms) between the moment the first INVITE is sent in the originate leg and the moment a provisional response/final response comes from the SIP device.  It is calculated by the Billing Engine as the time difference between <i>h323-setup-time</i> and <i>alert-timepoint</i> ( <i>h323-connect-time</i> is used instead of <i>alert-timepoint</i> , if the latter is not present in the accounting request)
<i>connect_time</i>	dateTime	The time when the session started (e.g., the call got connected)
<i>disconnect_time</i>	dateTime	The time when the session was disconnected or was last updated (for incremental xDR charge)
<i>bill_time</i>	dateTime	Call bill time
<i>disconnect_cause</i>	int	Specifies the reason the call was disconnected. This is a hexadecimal code
<i>voice_quality</i>	int	Shows call quality provided during the session as/if reported by NAS (provided for voice calls by Cisco equipment)
<i>used_quantity</i>	int	The quantity that was charged in original measurement units (e.g. duration of the voice call in seconds)
<i>account_id</i>	string	ID of the account being charged
<i>i_account</i>	int	The unique ID of the account's record in the database
<i>i_dest</i>	int	The unique ID of the destination record
<i>i_dest_group</i>	int	The unique ID of the destination group
<i>i_tariff</i>	int	The unique ID of the tariff record
<i>i_rate</i>	int	The unique ID of the rate record

<i>charged_quantity</i>	int	The number of charged units (the service is measured based on <i>used_quantity</i> after applying rounding intervals)
<i>charged_amount</i>	float	The amount charged in the respective currency
<i>history</i>	string	The formula that shows charge application steps or a textual description of the charge applied
<i>peak_level</i>	int	The ID of the peak level: 0 (peak period), 1 (off-peak period), 2 (2nd off-peak period)
<i>i_vendor</i>	int	Applicable for vendor xDRs. The unique ID of the Vendor record who was billed
<i>i_service</i>	int	The unique ID of the service record
<i>i_accessibility</i>	int	The unique ID of the rating entry record in the database
<i>i_customer</i>	int	The unique ID of the customer or the reseller (where applicable) associated with the call. This ID should be equal to the corresponding account's owner in CDR_Accounts, to the xDR owner in CDR_Customers and to the direct customer in CDR_Vendors
<i>h323_incoming_conf_id</i> ( <i>parent_session_id</i> )	string	The unique ID of the incoming session (if exists) used for interrelating xDRs, when the charged session is established as a result of a previous session (possibly having its own xDR)
<i>rating_pattern</i>	string	The rate pattern that was used to match a destination in the tariff rates
<i>subscriber_ip</i>	string	The internet address of the end-user (if applicable) which was assigned to him in the moment of charge application
<i>split_order</i>	int	The xDR order number in case it is a part of the group of xDRs created for the same session
<i>billing_model</i>	int	Specifies the account type; one of the following: -1 – Debit account 1 – Credit account



---

<i>i_invoice</i>	int	The unique ID of the invoice record. Designates the invoice where the xDR was settled
<i>revenue</i>	float	Applicable for vendor xDRs. Shows total revenue collected for the related session