



**DEVELOPER GUIDE**

August 2018 | 3725-49022-002A

# **Polycom® Zero Touch Provisioning API**



Copyright© 2018, Polycom, Inc. All rights reserved. No part of this document may be reproduced, translated into another language or format, or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Polycom, Inc.

6001 America Center Drive  
San Jose, CA 95002  
USA

### **Trademarks**

Polycom®, the Polycom logo and the names and marks associated with Polycom products are trademarks and/or service marks of Polycom, Inc. and are registered and/or common law marks in the United States and various other countries.



All other trademarks are property of their respective owners. No portion hereof may be reproduced or transmitted in any form or by any means, for any purpose other than the recipient's personal use, without the express written permission of Polycom.

### **Disclaimer**

While Polycom uses reasonable efforts to include accurate and up-to-date information in this document, Polycom makes no warranties or representations as to its accuracy. Polycom assumes no liability or responsibility for any typographical or other errors or omissions in the content of this document.

### **Limitation of Liability**

Polycom and/or its respective suppliers make no representations about the suitability of the information contained in this document for any purpose. Information is provided "as is" without warranty of any kind and is subject to change without notice. The entire risk arising out of its use remains with the recipient. In no event shall Polycom and/or its respective suppliers be liable for any direct, consequential, incidental, special, punitive or other damages whatsoever (including without limitation, damages for loss of business profits, business interruption, or loss of business information), even if Polycom has been advised of the possibility of such damages.

### **End User License Agreement**

BY USING THIS PRODUCT, YOU ARE AGREEING TO THE TERMS OF THE END USER LICENSE AGREEMENT (EULA) AT: <http://documents.polycom.com/indexes/licenses>. IF YOU DO NOT AGREE TO THE TERMS OF THE EULA, DO NOT USE THE PRODUCT, AND YOU MAY RETURN IT IN THE ORIGINAL PACKAGING TO THE SELLER FROM WHOM YOU PURCHASED THE PRODUCT.

### **Patent Information**

The accompanying product may be protected by one or more U.S. and foreign patents and/or pending patent applications held by Polycom, Inc.

### **Open Source Software Used in this Product**

This product may contain open source software. You may receive the open source software from Polycom up to three (3) years after the distribution date of the applicable product or software at a charge not greater than the cost to Polycom of shipping or distributing the software to you. To receive software information, as well as the open source software code used in this product, contact Polycom by email at [OpenSourceVideo@polycom.com](mailto:OpenSourceVideo@polycom.com).

### **Customer Feedback**

We are striving to improve our documentation quality and we appreciate your feedback. Email your opinions and comments to [DocumentationFeedback@polycom.com](mailto:DocumentationFeedback@polycom.com).

### **Polycom Support**

Visit the [Polycom Support Center](#) for End User License Agreements, software downloads, product documents, product licenses, troubleshooting tips, service requests, and more.

# Contents

---

<b>About This Guide .....</b>	<b>4</b>
Audience and Purpose.....	4
Get Help .....	4
<i>Related Documents</i> .....	4
<i>Polycom and Partner Resources</i> .....	4
<i>The Polycom Community</i> .....	4
<b>Using the Zero Touch Provisioning API .....</b>	<b>5</b>
Before You Begin .....	5
<i>Create an API Account</i> .....	5
<i>Create a Configuration Profile in the ZTP Console</i> .....	5
ZTP API Address .....	5
Associate a Device Using the API .....	5
Additional API Operations.....	6
<b>API Command Descriptions .....</b>	<b>8</b>
add-package .....	8
add-sip-device.....	9
copy-sip-device .....	10
create-subscriber .....	11
delete-sip-device .....	12
modify-sip-device .....	14
query-accountid-by-mac .....	16
query-sip-device.....	17
set-location.....	18
swap-sip-device .....	19
<b>Appendix A: Sample API .....</b>	<b>21</b>
ZTP API Sample File Arguments.....	21
Deploy the ZTP API Sample File .....	22
Use the ZTP API Test Utility .....	23

# About This Guide

---

This guide explains the Application Program Interface (API) used to automate profile associations within the Polycom® Zero Touch Provisioning (ZTP) solution.

This document includes the following:

- An overview of device associations you can perform with the API
- A detailed description of the API commands
- A sample HTML routine that illustrates the API commands

## Audience and Purpose

This guide is intended for administrators and Polycom channel partners who want to use the ZTP solution to create and manage profile associations or integrate the ZTP solution into their existing provisioning process.

## Get Help

For more information about installing, configuring, and administering Polycom products, refer to Documents and Downloads at [Polycom Support](#).

## *Related Documents*

The following documentation is available on [Polycom Support](#):

- *Polycom Zero Touch Provisioning Provisioning Guide*
- *Polycom Zero Touch Provisioning Frequently Asked Questions*

## *Polycom and Partner Resources*

To find all Polycom partner solutions, see [Strategic Global Partner Solutions](#).

## *The Polycom Community*

The [Polycom Community](#) gives you access to the latest developer and support information. Participate in discussion forums to share ideas and solve problems with your colleagues. To register with the Polycom Community, simply create a Polycom online account. When logged in, you can access Polycom support personnel and participate in developer and support forums to find the latest information on hardware, software, and partner solutions topics.

# Using the Zero Touch Provisioning API

---

This section explains how to associate your devices using the Zero Touch Provisioning (ZTP) API.



You can use a mixture of the ZTP console and the API interface to associate devices. For example, you can use the console to create the Reference ID (account-id) and the API to create device associations against that Reference ID (account-id).

## Before You Begin

Before you can associate devices using the ZTP API, you must first set up an API account and create a configuration profile for your device.

### Create an API Account

You must create an API account separate from your ZTP Web Console account to use the ZTP API. When you write the APIs in XML, you enter your API account credentials in each API message header.

#### To create an API Account:

- » Email [ztpinfo@polycom.com](mailto:ztpinfo@polycom.com) to register for the account and obtain your credentials.

### Create a Configuration Profile in the ZTP Console

You first must create a configuration profile in the ZTP console to use the API to associate a device. You can then create API commands to perform automated profile associations within ZTP. Note that the API commands cover a broader set of capability than is required for the ZTP solution.

For information on creating and managing configuration profiles, see the *Polycom Zero Touch Provisioning Provisioning Guide* at Polycom Support.



You cannot use the ZTP API to create or modify configuration profiles.

## ZTP API Address

The ZTP API address is: <https://ztpconsole.polycom.com/inboundservlet/GenericServlet>

## Associate a Device Using the API

This section explains how to associate a device using the API.

#### To perform device associations:

- 1 Create a **Reference ID** (account-id) using the following API commands:

- [create-subscriber](#) Creates a unique identifier within the ZTP system. This command includes the following fields and elements:
  - ◆ `account-id` Displays in the ZTP web console as the **Reference ID** and tracks associations in the ZTP system.  
The value you enter for the `account-id` must be unique. The API will return an error if you choose a value that is already in use, even if created by another ZTP user.
  - ◆ `isp-name` Polycom provides this information when you obtain an API account.  
This field is case sensitive.
  - ◆ *PersonalInformation* Optional element used to include personal or business information.  
You can include contact information such as name and address or the location to which the devices are being shipped.
- [set-location](#) Required for each **Reference ID** (`account-id`). This command must include the following field options:
  - ◆ `account-id` Same value used in `create-subscriber`.
  - ◆ `location-name` Set to **default**.
- [add-package](#) Required for each **Reference ID** (`account-id`). This command must include the following field options:
  - ◆ `account-id` Same value used in `create-subscriber`.
  - ◆ `base-package-name` Set to **default**.

## 2 Use the [add-sip-device](#) API command to associate the profile.

The `account-id` for each device association must be the same value used in `create-subscriber`.



Each API command generates a message response from ZTP. Polycom recommends that you check this message response using the `query-sip-device` command to confirm that an operation has been successfully completed.

## Additional API Operations

In addition to associating devices, you can use API commands to perform the following operations:

- [copy-sip-device](#) Copy a device association from one device to another.
- [swap-sip-device](#) Swap device associations between two devices.
- [delete-sip-device](#) Delete a device association.
- [modify-sip-device](#) Modify a device association. When modifying device associations, you must use the same credentials (user ID and password) and the same value for the `account-id` field you used to create the device association.



You can use the ZTP console to find device associations to confirm that they have been correctly performed.

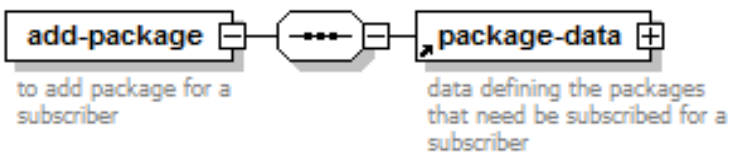
# API Command Descriptions

This section explains the API commands you can use and includes examples of the XML schema that defines the interface. See [Appendix A: Sample API](#) at the end of this guide for an example API.

## add-package

The `add-package` API command consists services of the subscriber account.

Although there is only one default value, you must set the `package-data` element value to **default**.



## Elements

- `package-data` Data defining the packages that need to be subscribed for a subscriber.

## Attributes

- `account-id`
- `base-package-name` Required string. Always set to **default**.

## Example

The following table illustrates an example XML API for adding a package.

```
<?xml version='1.0' encoding='UTF-8'?>
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\GenericInbound\schema\Generic-inbound-V1.1.xsd' userid='special-login'
password='some-password' message-id='111'>
<add-package account-id='myaccount'>
<package-data>
<base-package-name>default</base-package-name>
</package-data>
</add-package>
</request>

-----
<?xml version='1.0'?>
<response message-id='111' xmlns='http://schemas.alopa.com/Inbound'>
  <status ErrorCode='0000' ErrorMessage='Successful.' />
</response>
```



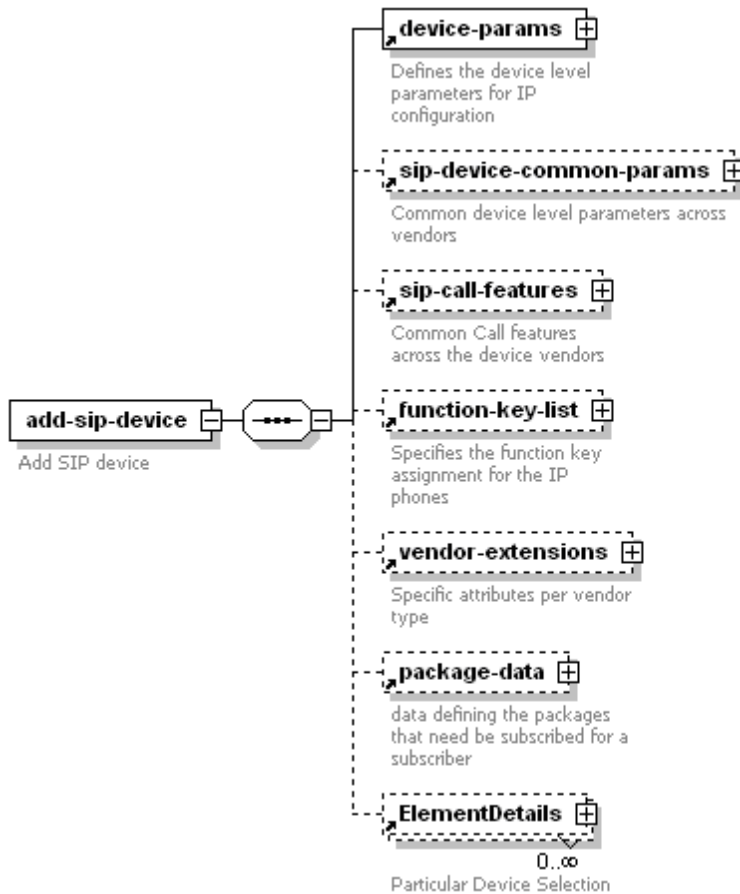
```

<response-details>
  <standard-response accountid='myaccount' />
</response-details>
</response>

```

## add-sip-device

Use the `add-sip-device` API command to provision a device to ZTP.



## Elements

- *device-params* Defines the device level parameters for IP configuration.
- *sip-device-common-params* Common device-level parameters across vendors.
- *sip-call-features* Common call features across the device vendors.
- *function-key-list* Specifies the function key assignment for the IP phones.
- *vendor-extensions* Specific attributes per vendor type.
- *package-data* Data defining the packages that need to be subscribed for a subscriber.
- *ElementDetails* Particular device selection.

## Attributes

- `account-id` Required string.
- `vendor` Required string. Always set to **Polycom**.
- `vendorModel` Required string. Always set to **Polycom\_UCS\_Device**.
- `templateCriteria` Required string. Set to the profile created by the service provider.

## Example

The following table illustrates an example XML API for adding a SIP device.

```
<request xmlns="http://schemas.alopa.com/Inbound"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.alopa.com/Inbound
C:\GenericInbound\schema\Generic-inbound-V1.1.xsd" userid="special-login"
password="some-password" message-id="1001">
<add-sip-device account-id="myaccount">
<device-params><deviceId>0000cafe8897</deviceId>
<serialNo>0000cafe8897</serialNo>
<vendor>Polycom</vendor>
<vendorModel>Polycom_UCS_Device</vendorModel>
</device-params>
<sip-device-common-params>
<templateCriteria>yourProfile</templateCriteria>
</sip-device-common-params>
<package-data><base-package-name>default</base-package-name>
</package-data></add-sip-device></request>
```

## copy-sip-device

Use the `copy-sip-device` API command to copy a profile to provision new devices with the same features as an existing device associated with an account ID.

**copy-sip-device**

clone a sip device

## Elements

- `device-params` Defines the device level parameters for phone/device configuration.

## Example

The following table illustrates an example XML API for copying a profile.

```

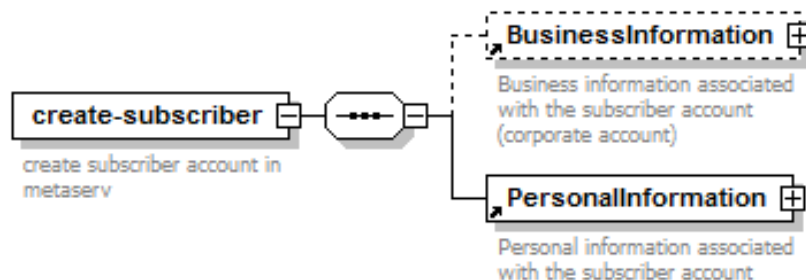
<?xml version='1.0' encoding='UTF-8'?>
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\GenericInbound\schema\Generic-inbound-V1.1.xsd'
userid='special-login' password='some-password' >
<copy-sip-device old-mac='343435353535' newmac='
089282928928' >
</copy-sip-device>
</request>
-----
<?xml version='1.0'?>
<response xmlns='http://schemas.alopa.com/Inbound'>
<status ErrorCode='0000' ErrorMessage='Successful.' />
<response-details>
<standard-response accountid='100212' />
</response-details>
</response>

```

## create-subscriber

Use the `create-subscriber` API command to create an account for the subscriber in ZTP that includes the subscriber's personal and business information.

The API provides business information that is not stored in the ZTP database. If you need to store business information to the ZTP database, you must modify the corresponding business flow in ZTP.



## Elements

- *BusinessInformation* Business information associated with the subscriber account (corporate account). This is an optional element that is currently not stored in ZTP.
- *PersonallInformation* Personal information associated with the subscriber account.

## Attributes

- `account-id` Required string. This value can be an order number for a unique customer reference number, but it must be unique across ZTP.

- `isp-name` Required string. This value is the company ID associated with the user and is given at the time of account creation. This string is case sensitive.

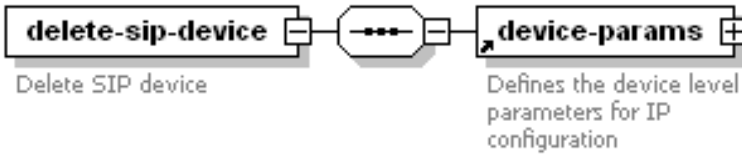
## Example

The following table illustrates an example of XML API for creating a subscriber:

```
<?xml version='1.0' encoding='UTF-8'?>
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\GenericInbound\schema\Generic-inbound-V1.1.xsd' userid='special-login'
password='some-password' message-id='1001' >
<create-subscriber account-id = 'myaccount' isp-name= 'companyid'>
<PersonalInformation>
<FirstName>Sharada</FirstName>
<LastName>Pappu</LastName>
<password>user1223</password>
<address>
<StreetAddress1>vani</StreetAddress1>
<StreetAddress2>vilas</StreetAddress2>
<City>Santa Clara</City>
<State>KA</State>
<Zipcode>560004</Zipcode>
<Country>India</Country>
</address>
<phone>6618004</phone>
<select-location></select-location>
<cmmac></cmmac>
</PersonalInformation>
</create-subscriber>
</request>
-----
<?xml version='1.0'?>
<response message-id='1001' xmlns='http://schemas.alopa.com/Inbound'>
  <status ErrorCode='0000' ErrorMessage='Successful.' />
  <response-details>
    <standard-response accountid='myaccount' />
  </response-details>
</response>
```

## delete-sip-device

The `delete-sip-device` API command deletes SIP service for a subscriber.



## Elements

- *device-params* Defines the device level parameters for IP configuration.

## Attributes

- `account-id` Required string.
- The subscriber user ID or account number in ZTP.

## Example

The following table illustrates an example XML API for deleting a device:

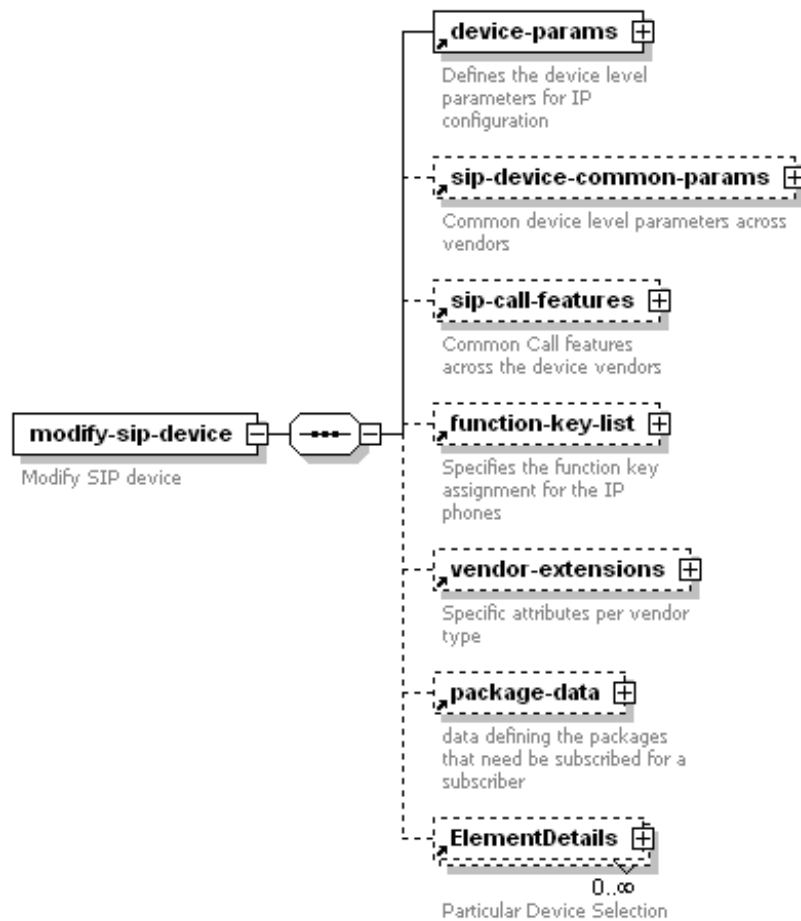
```
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' userid='special-login' password='some-
password' message-id='String' model-name='String'>
<delete-sip-device xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' accountid='test-sip'>
<device-params>
</device-params>
</delete-sip-device>
</request>
-----
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<response xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd'>
<status ErrorCode='String' ErrorMessage='Successful.' />
<response-details>
</response-details>
</response>
```

## modify-sip-device

The `modify-sip-device` API command modifies subscriber SIP device parameters, including the registration address or proxy address.

You cannot use this API to modify a package.

You need to match the `account id` and the MAC address of the SIP device for this operation to be successful.



### Elements

- *device-params* Defines the device level parameters for IP configuration.
- *sip-device-common-params* Common device-level parameters across vendors.
- *sip-call-features* Common call features across the device vendors.
- *function-key-list* Specifies the function key assignment for the IP phones.
- *vendor-extensions* Specific attributes per vendor type.
- *package-data* Data defining the packages that need to be subscribed for a subscriber.
- *ElementDetails* Particular device selection.

## Attributes

- account-id Required string.
- The subscriber user ID or account number in ZTP.

## Example

The following table illustrates an example XML API for copying a profile:

```
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U http://www.xmlspy.com)-->
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' userid='special-login' password='some-
password' message-id='String' model-name='String'>
<add-computer accountid='String'>
<modify-sip-device xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' accountid='test-sip'>
<device-params>
</device-params>
<sip-device-common-params>
</sip-device-common-params>
<sip-call-features>
</sip-call-features>
<function-key-list>
<key number='10'>
</key>
</function-key-list>
<vendor-extensions/>
<package-data>
<base-package-name>Sample_SIP</base-package-name>
<count>0</count>
</package-data>
</modify-sip-device>
</request>
-----
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<response xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd'>
<status ErrorCode='String' ErrorMessage='Successful.' />
<response-details>
</response-details>
</response>
```

## query-accountid-by-mac

Use the `query-accountid-by-mac` API command to query an account ID using the MAC address of a device.

### query-accountid-by-mac

Return Account-id given  
mac-address

## Elements

- `device-params` Defines the device level parameters for IP configuration.

## Attributes

- `mac-address` Required string.
- `device type` Optional string. You can query any of the following device types:
  - CM
  - SIP
  - MTA
  - Generic

## Example

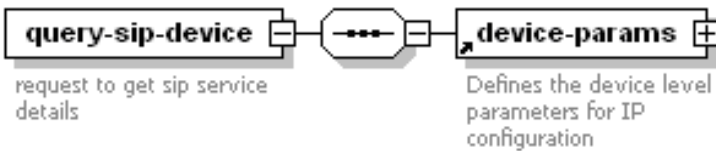
The following table illustrates an example XML API for querying an account ID by MAC address.

```
<?xml version='1.0' encoding='UTF-8'?>
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\GenericInbound\schema\Generic-inbound-V1.1.xsd' userid='special-login'
password='some-password' >
<query-accountid-by-mac mac-address='289282928928'>
</query-accountid-by-mac>
</request>
-----
<?xml version='1.0'?>
<response xmlns='http://schemas.alopa.com/Inbound'>
  <status ErrorCode='0000' ErrorMessage='Successful.' />
  <response-details>
    <standard-response accountid='10023' />
  </response-details>
</response>
```



## query-sip-device

Use the `query-sip-device` API to query an existing SIP device for an account ID.



### Elements

- `device-params` Defines the device level parameters for IP configuration.

### Attributes

- `account-id` Required string.
- The subscriber user ID or account number in ZTP.

### Example

The following table illustrates an example XML API for querying a device.

```
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' userid='special-login' password='some-
password' message-id='String' model-name='String'>
<query-sip-device xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' accountid='test-sip'>
<device-params>
</device-params>
</query-sip-device>
</request>
-----
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<response message-id='1001'>
<status ErrorCode='0000' ErrorMessage='Successful' />
<response-details>
<sip-device-response>
<sip-call-features>
<feature>
</feature>
<feature>
```

```

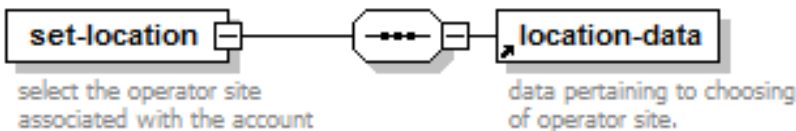
</feature>
</sip-call-features>
<vendor-extensions>
</vendor-extensions>
<line-details>
<line port='1' sipURI='sigma@sigmafoft.biz' >
<sip-line-params>
</sip-line-params>
<sip-call-features>
<feature>
</feature>
</sip-call-features>
<vendor-extensions>
<type>shared</type>
</vendor-extensions>
</line>
</line-details>
</sip-device-response>
</response-details>
</response>

```

## set-location

Use the `set-location` API command to add an operator site, which is the location where the subscriber's package is available in deployment. If you add a location where the package is not available, the operation will complete; however, subsequent operations for that subscriber will fail.

Although there is only one default value, you must set the `location-name` value to **default**.



## Elements

- `location-data` Data pertaining to choosing the operator site.

## Attributes

- `account-id` Required string.
- `location-name` Required string. Always set to **default**.

## Example

The following table illustrates an example XML API for adding a location:

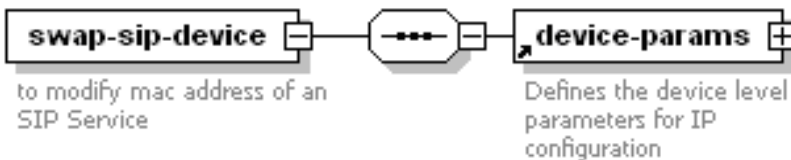
```
<?xml version='1.0' encoding='UTF-8'?>
<request userid='special-login' password='some-password' message-id='MESSAGE_ID'>
<set-location account-id='myaccount'>
<location-data location-name='default' />
</set-location>
</request>

-----

<?xml version='1.0'?>
<response message-id='MESSAGE_ID' xmlns='http://schemas.alopa.com/Inbound'>
  <status ErrorCode='0000' ErrorMessage='Successful.' />
  <response-details>
    <standard-response accountid='nasa001' />
  </response-details>
</response>
```

## swap-sip-device

Use the `swap-sip-device` API command to swap the MAC address of an existing SIP device with the MAC address of a new SIP device.



## Elements

- *device-params* Defines the device level parameters for IP configuration.

## Attributes

- `account-id` Required string.
- The subscriber user ID or account number in ZTP.
- `newDeviceId` Required string.
- The MAC address of the new SIP device.

## Example

The following table illustrates an example XML API for swapping a device.

```
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
```

```
<request xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' userid='special-login' password='some-
password' message-id='String' model-name='String'>
<swap-sip-device xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd' newDeviceId='0000cafebabe'
accountid='test-sip'>
<device-params>
</device-params>
</swap-sip-device>
</request>
-----
<?xml version='1.0' encoding='UTF-8'?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<response xmlns='http://schemas.alopa.com/Inbound'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://schemas.alopa.com/Inbound
C:\DOCUME~1\Desktop\Generic-inbound-V1.1.xsd'>
<status ErrorCode='String' ErrorMessage='Successful.' />
<response-details>
</response-details>
</response>
```

# Appendix A: Sample API

---

This appendix illustrates an API that performs the operations outlined in this guide. The ZTP API Sample File gives an example of how the API can be used to perform operations within ZTP. When deployed, the ZTP API Sample File creates the ZTP API Test Utility.

- **Usage:** `plcm_ztp_tool.rb [arguments]`
- **Example:** `plcm_ztp_tool.rb -s ztpconsole.polycom.com -u inbound -p inbound123 -a testusers -n 10`



The contents of the ZTP API Sample File are provided as is without warranty. You should not use this file or its contents for production purposes without adequate review and testing.

## ZTP API Sample File Arguments

The following arguments are used in the ZTP API Sample File.

### Mandatory Arguments:

<i>Argument</i>	<i>Description</i>
<code>-u, --user USERID</code>	Inbound user ID for authenticating to the Device Provisioning Manager.
<code>-p, --password PASSWORD</code>	Inbound password for authenticating to the Device Provisioning Manager.
<code>-s, --server HOST</code>	Hostname of the Device Provisioning Manager server.
<code>-a, --account-prefix PREFIX</code>	String to prefix to new subscriber account ID values. Example: A prefix value of "test" will generate account IDs of "test001", "test002", "test003" and so on.
<code>-n, --num NUM</code>	Number of subscribers to create.

### Optional Arguments:

<i>Argument</i>	<i>Description</i>
<code>-l, --location [LOCATION]</code>	Operator site location name to associate with each new subscriber. (Optional) Default: "default"
<code>-k, --package [PACKAGE]</code>	Service package name to associate with each new subscriber. (Optional) Default: "default"

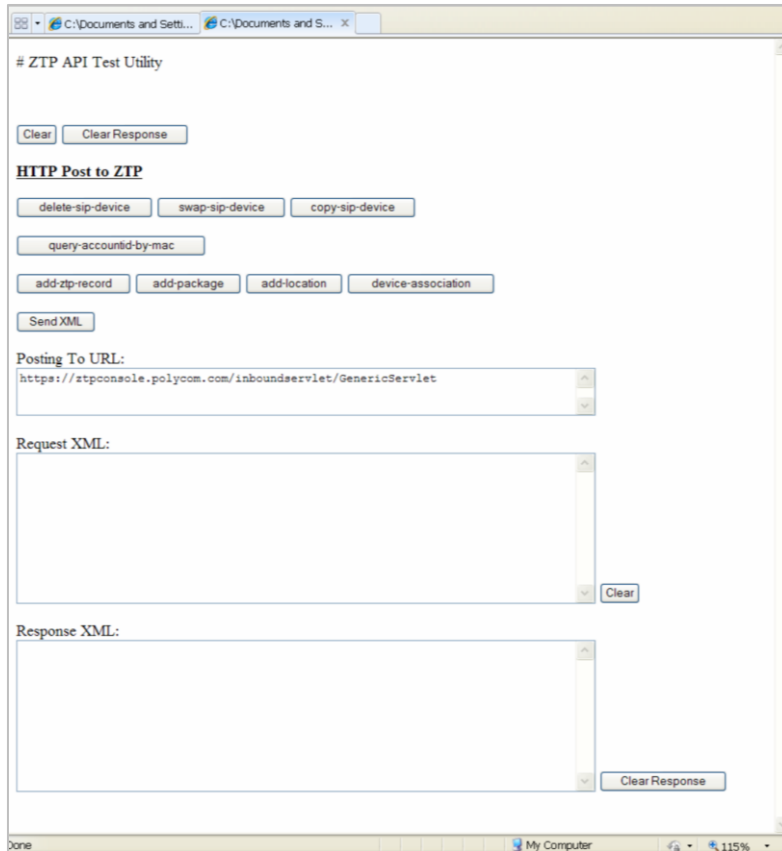
<i>Argument</i>	<i>Description</i>
<code>-i, --isp-name [ISP-NAME]</code>	ISP name to set for each new subscriber. (Optional) Default: "pete_comp"
<code>-d, --devices [NUM]</code>	Number of SIP devices to add to each new subscriber. (Optional) Default: "1"
<code>-v, --vendor [VENDOR]</code>	Vendor name to set for each new SIP device. (Optional) Default: "Polycom"
<code>-m, --model [MODEL]</code>	Vendor model name to set for each new SIP device. (Optional) Default: "Polycom_UCS_Device"
<code>--no-ssl</code>	To use SSL/TLS when connecting to the Device Provisioning Manager. (Optional) Default: Connect using SSL/TLS
<code>-h, --help</code>	Display this screen.

## Deploy the ZTP API Sample File

You can deploy the ZTP API Sample File using your specific account information.

### To deploy the ZTP API Sample File:

- 1 Open the file with a text or HTML editor and do the following:
  - a Replace all occurrences of ***myloginname*** with the login name for your API account.
  - b Replace all occurrences of ***mypassword*** with the password for your API account.
  - c Replace all occurrences of `isp-name` with the `isp-name` for your ZTP access.
- 2 Save the file.
- 3 Open the file using a web browser. A screen entitled *# ZTP API Test Utility* displays.



## Use the ZTP API Test Utility

You can use the ZTP API Test Utility to perform the operations as outlined in this guide.

### To use the ZTP API Test Utility:

- 1 Click the button for the API operation you wish to perform.  
For example, to create a **Reference ID** (`account-id`), click the following options:
  - `add-ztp-record`
  - `add-package`
  - `add-location`

The **Request XML** field populates with the correctly formatted XML text for the selected operation.

- 2 Modify the default XML text in the **Request XML** field as needed.
- 3 Click **Send XML** to send the XML request to the ZTP server.  
The response displays in the **Response XML** field.



This Utility and the .xsd are available along with access to the API through [ztpinfo@polycom.com](mailto:ztpinfo@polycom.com)