

Import User and Device Data from CSV Files

Introduction

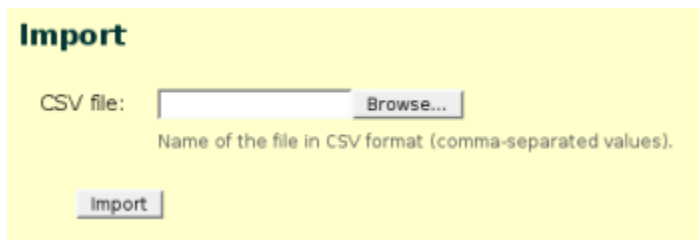
sipXconfig 3.1/3.2 provides two different mechanisms to import external data.

1. The sipX [SOAP interface](#) is a sophisticated Web services based interface that offers a full ConfigServer API.
2. CSV import is a simple mechanism where as part of the deployment planning process you create a "cut-sheet" using your favorite spreadsheet application. Data for users and devices can then be imported into ConfigServer using the CSV file format.

CSV File Import Functionality

In many cases, especially during an initial install, what you are trying to do is fairly simple: You create a list of users and their respective devices (one line per device) in a spreadsheet application as part of your dial plan planning process. You then want to easily upload that data into ConfigServer avoiding the need to re-type everything. This is exactly what the CVS import functionality offers you.

The idea is that you prepare the file in your favorite spreadsheet application (Open Office Calc, Excel, gnumeric etc.) and export it to a .csv file. Later you can upload the file and sipXconfig will create the users and phones for you. You can put phones in one or more groups, which helps if you want to configure common parameters for them.



v4.2 The [CSV file](#) has to have a title line (though the actual title strings don't matter) and the following columns (in the listed order):

- User name
- Voice-mail PIN
- SIP password
- First name
- Last name
- User alias
- Email
- User group
- Phone serial number
- Phone model
- Phone group
- Phone description

(NOTE: use this template [cutsheet.csv](#) for sipXconfig Ver.4.2 and older)

Each line from the imported file will result in the creation of a user with the phone assigned to that user (One line per device). If *user group* or *phone group* fields are not empty, the newly created user and phone will be added to the respective group. Groups will be created if they do not exist already.

If a user with the same username is already present, sipXconfig will update the existing user instead of creating a new one. The same is true for phones: If the phone with an identical serial number already exist it will be updated. Please note that if the phone exists already its model will not be changed.

Only the user name and the phone serial number are mandatory fields. You can leave the remaining fields empty, in which case sipXconfig will not overwrite their values. However, the column layout of the spreadsheet file has to be maintained.

v4.4 Version 4.4 added new columns in the CSV file as follows (in the listed order):

- User name
- Voice-mail PIN
- SIP password
- First name
- Last name
- User alias
- EMail address
- User group
- Phone serial number
- Phone model
- Phone group
- Phone description
- Im Id

- Job Title
- Job department
- Company name
- Assistant name
- Cell phone number
- Home phone number
- Assistant phone number
- Fax number
- Alternate email
- Alternate im
- Location
- Home street
- Home city
- Home state
- Home country
- Home zip
- Office street
- Office city
- Office state
- Office country
- Office zip
- Did number
- Office mail stop
- Active greeting - it has one of the following values: "none", "standard", "outoffice", "extendedabsence" which means Default system greeting, Standard, Out of office and Extended absence
- Email voicemail notification - it has one of the following values: "0", "1", "2" which means 'No notification', 'E-mail notification' and 'Synchronize with the IMAP server'
- Email format - it has one of the following values: "FULL", "MEDIUM", "BRIEF"
- Email attach audio - it has one of the following values: "true"/"false"
- Alternate email voicemail notification - it has one of the following values: "0", "1", "2" which means 'No notification', 'E-mail notification' and 'Synchronize with the IMAP server'
- Alternate email format - it has one of the following values: FULL, MEDIUM, BRIEF
- Alternate email attach audio - it has one of the following values: "true"/"false"
- Internal Voicemail Server - it has one of the following values: "true"/"false" which means Internal Voicemail Server/Microsoft Exchange UM Voicemail Server
- Caller ID - externally visible user Caller ID
- Block Caller ID - it has one of the following values: "true"/"false"
- Additional phone settings - it contains MicroBrowser settings for Polycom phones and it has the following form: "setting_name1=setting_value1, setting_name2=setting_value2...".
The following settings are supported: mb/proxy, mb/idleDisplay/home, mb/idleDisplay/refresh, mb/main/home, mb/main/idleTimeout, mb/main/statusbar, mb/limits/nodes, mb/limits/cache
- Additional line settings - it contains the text label which appear on the Polycom's display adjacent to the associated line key and it has the following form: "reg/label=your_text_label"

v4.6 Version 4.6 added new columns in the CSV file as follows (in the listed order):

- User name
- PIN
- Voicemail PIN
- SIP password
- First name
- Last name
- User alias
- EMail address
- User group
- Phone serial number
- Phone model
- Phone group
- Phone description
- Im Id
- Salutation
- Manager
- Employeeid
- Job Title
- Job department
- Company name
- Assistant name
- Cell phone number
- Home phone number
- Assistant phone number
- Fax number
- Did number
- Alternate email
- Alternate im
- Location
- Home street
- Home city
- Home state

- Home country
- Home zip
- Office street
- Office city
- Office state
- Office country
- Office zip
- Office mail stop
- Twitter
- LinkedIn
- Facebook
- Xing
- Active greeting - it has one of the following values: "none", "standard", "outofoffice", "extendedabsence" which means Default system greeting, Standard, Out of office and Extended absence
- Email voicemail notification - it has one of the following values: "0"/"1" which means 'No notification'/'E-mail notification'
- Email format - it has one of the following values: "FULL", "MEDIUM", "BRIEF"
- Email attach audio - it has one of the following values: "true"/"false"
- Alternate email voicemail notification - it has one of the following values: "0"/"1" which means 'No notification'/'E-mail notification'
- Alternate email format - it has one of the following values: FULL, MEDIUM, BRIEF
- Alternate email attach audio - it has one of the following values: "true"/"false"
- Internal Voicemail Server - it has one of the following values: "true"/"false" which means Internal Voicemail Server/Microsoft Exchange UM Voicemail Server
- Caller ID - externally visible user Caller ID
- Block Caller ID - it has one of the following values: "true"/"false"
- Additional phone settings - it contains MicroBrowser settings for Polycom phones and it has the following form: "setting_name1=setting_value1, setting_name2=setting_value2,..".
The following settings are supported: mb/proxy, mb/idleDisplay/home, mb/idleDisplay/refresh, mb/main/home, mb/main/idleTimeout, mb/main/statusbar, mb/limits/nodes, mb/limits/cache
- Additional line settings - it contains the text label which appear on the Polycom's display adjacent to the associated line key and it has the following form: "reg/label=your_text_label"

Valid Manufacturer / Model Combinations

{{Box Important|Manufacturer and Model names are case sensitive - make sure that you use values listed in the table below.}}

Manufacturer	Model
polycom	300
polycom	430
polycom	500
polycom	600
polycom	4000
ciscosp	7960
ciscosp	7940
ciscoAta	18x
ciscoAta	7905
ciscoAta	7912
grandstream	PhoneBt
grandstream	PhoneGxp
grandstream	Ht286
grandstream	Ht386
grandstream	Ht486
grandstream	Ht488
grandstream	Ht496
snom	360
snom	320

v3.8

CSV import for phone models going forward is only one field. This list is not meant to be inclusive, if you know web interface has support for the phone model you need, try to extrapolate the modelId from this list (e.g. polycom9000).

- polycom300
- polycom430
- polycom500
- polycom550
- polycom650
- polycom600
- polycom4000
- cisco7960
- cisco7940
- cisco18x
- cisco7905
- cisco7912
- gsPhoneBt
- gsPhoneGxp
- gsPhoneGxv3000
- gsHt286
- gsHt386
- gsHt486
- gsHt488
- gsHt496
- snom300
- snom320
- snom360

Or you can use ConfigServer to create a phone of the right type, then from the shell use these commands to see what its modelId is:

```
$ psql -U postgres SIPXCONFIG
Welcome to psql 8.1.9, the PostgreSQL interactive terminal.
```

Type: \copyright for distribution terms

```
\h for help with SQL commands
? for help with psql commands
\g or terminate with semicolon to execute query
\q to quit
```

SIPXCONFIG=# select serial_number, model_id from phone;

```
serial_number | model_id
-----+-----
0004f203c49a | polycom600
0004f202ec81 | polycom300
00405a187c1a | lip6830
00405a187cb2 | lip6830
0004f211cead | polycom550
(5 rows)
```

SIPXCONFIG=# \q

```
$
```