

Zoiper provisioning documentation

How to setup provisioning of the configuration of Zoiper5

- *Product release:* **Zoiper5**
- *Updated:* **September 2019**

Contents

- [Zoiper provisioning documentation](#)
- [Contents](#)
 - [Overview](#)
 - [What is provisioning?](#)
 - [Who needs provisioning?](#)
 - [In which versions of Zoiper is provisioning available?](#)
 - [Benefits of provisioning](#)
 - [How does provisioning work?](#)
 - [Provisioning configuration](#)
 - [Structure of the provisioning configuration file](#)
 - [The **options** section of the provisioning configuration file](#)
 - [How does provisioning of the configuration file work?](#)
 - [Error response from the provisioning server](#)
 - [Example contents of the provisioning configuration file](#)

Overview

What is provisioning?

Provisioning is an easy way to remotely setup the local (i.e. end-user) configuration of Zoiper.

Who needs provisioning?

Provisioning is particularly useful for VoIP service providers and call centers.

In which versions of Zoiper is provisioning available?

Provisioning is only available in the **Premium** versions of *Zoiper5*.

Benefits of provisioning

- Relieve your users from the burden of having to configure various Zoiper settings, like accounts, codecs, etc.
- Load-balance (distribute the load of) your VoIP servers easily, without the necessity for manually updating the locally-stored server credentials.
- Use the credentials for your website instead of the username and password for the VoIP server (i.e. it is possible for them to be separate ones).

- Update the local configuration of your users in accordance with your specific requirements without involving them into any action on their behalf. This means that users might not even realize that changes in the configuration have taken place.

How does provisioning work?

Provisioning in *Zoiper5* is controlled by a special provisioning configuration file which is separate from the regular Zoiper configuration file. This file contains settings like the location of the provisioned (i.e. remote) configuration data, what authentication method is used, etc.

Provisioning takes place when Zoiper gets started. If the provisioning configuration file exists and is valid (i.e. is an XML document which conforms to the structure described below), Zoiper will download and apply the configuration from the provisioned configuration data using the parameters specified in the provisioning configuration file in accordance with the algorithm described below.

When provisioning begins, Zoiper first loads the data stored in the local configuration file and then overlays atop it the provisioned configuration data (which gets downloaded from the provisioning server for that purpose). Basically, this means that the values of the options from the local configuration data get replaced with their respective values in the provisioned configuration data, but only for values which are present in the provisioned configuration data (the values of the other ones do not get modified at all). This scheme allows for parts of the configuration to be defined only locally for each installation of Zoiper (like, for example, the audio options), and for others to be provisioned for every machine (like, for example, the user account options).

Provisioning configuration

The provisioning configuration file is named `provision.xml` and is an XML document that uses the *UTF-8* encoding. It should be located in one of the following locations (they get checked for a file named `provision.xml` in the described order):

1. The current working directory of the Zoiper process (i.e. the **Start In** field in shortcut properties on Windows).
2. The path to the Zoiper executable file:
 - On Windows, the executable is located in the Zoiper installation directory (i.e. it is a single one). By default Zoiper gets installed in `C:\Program Files (x86)\Zoiper5`.
 - On Linux, the executable is usually located in one of the directories designated for executable files (e.g. `/usr/bin`, `/usr/local/bin`, `/bin`, etc).
3. The Zoiper *application data directory* for the current user. By default it has one of the following paths, depending on the platform:
 - On Windows, the path to the directory is `%APPDATA%\Zoiper5`. This location gets interpreted in a different way for different Windows versions (here `USER` is a placeholder for the name of the current user):
 - on Windows versions before *Vista* it gets translated to `C:\Documents and Settings\USER\Application Data\Zoiper5`;
 - on *Windows Vista* and later it gets translated to `C:\Users\USER\AppData\Roaming\Zoiper5`.
 - On Linux, the path to the directory is `~/.Zoiper5`. It usually gets translated to `/home/USER/.Zoiper5`, where `USER` is a placeholder for the name of the current user.

This file is not included in the standard distribution of the Zoiper installation (i.e. it has to be created manually).

Structure of the provisioning configuration file

The provisioning configuration file is an XML document with the following structure:

- **provision**: the main configuration section (it corresponds to the root element of the document).
 - **base_urls**: a section with several sub-sections defining the base URLs used for provisioning, where each URL has its own **base_url** section.
 - **base_url**: this option defines a base URL used for provisioning.
 - If **url** authentication is used for the **authentication_type**, the user will be able to specify some parameter values for substitution in the URL. The currently available parameters are **[user]** and **[pass]**. They get substituted with one of the following things depending on the value of the **user_input_authentication** option:
 - the username and the password entered by the user, if **user_input_authentication** is enabled;
 - the **user** and **password** options, if **user_input_authentication** is disabled.
 - If provisioning using an URL fails, Zoiper will try to get provisioned using the next one. If it does not succeed with any of them, it will display an error message.
 - Type: **string**.
 - Default value: *nothing* (i.e. an empty string).
 - **options**: a section that contains the options which control the behavior of the provisioning process.

The **options** section of the provisioning configuration file

- **use_authentication**: this option determines whether authentication credentials will be provided to the webserver by Zoiper (i.e. whether authentication will be used at all).
 - Type: **boolean**.
 - Possible values: **false**, **true**.
 - Default value: **false**.
- **user_input_authentication**: this option determines whether Zoiper should ask the user to enter his authentication credentials (i.e. username and password). If the value of this option is **false**, the username and password from the **user** and **password** options are used for authentication.
 - Type: **boolean**.
 - Possible values: **false**, **true**.
 - Default value: **false**.
- **authentication_type**: this option determines the way Zoiper supplies the authentication credentials to the web server (i.e. how it gets *authenticated* to the server).
 - Type: **text enumeration**.
 - Possible values: **basic**, **url**:

- **basic**: basic access authentication is used for HTTP (for more information, see the relevant [Wikipedia article](#)).
 - **url**: the authentication credentials are passed as URL parameters. (See the **base_url** option for more information.)
 - Default value: **url**.
- **user**: this option defines a preset username which is used if **user_input_authentication** is **false**.
 - Type: **string**.
 - Default value: *nothing* (i.e. an empty string).
- **password**: this option defines a preset password which is used if **user_input_authentication** is **false**.
 - Type: **string**.
 - Default value: *nothing* (i.e. an empty string).
- **password_format**: this option determines the format in which the password is stored. It can either be MD5-encoded or a plaintext one.
 - Type: **enumeration**.
 - Possible values: **0, 1**:
 - **0**: means that the password will be *MD5*-encoded (i.e. the value of the **user** option will be a *MD5* hash of the password).
 - **1**: means that the password will be stored in plaintext (i.e. the value of the **user** option will be the actual unencrypted password).
 - Default value: **0**.
- **delete_accounts_beforehand**: this option determines whether the accounts present in the local configuration should get deleted before the provisioning happens (so that only the accounts from the provisioned configuration would be used).
 - Type: **boolean**.
 - Possible values: **false, true**.
 - Default value: **true**.
- **delete_contact_services_beforehand**: this option determines whether the contact services present in the local configuration should get deleted before the provisioning happens (so that only the contact services from the provisioned configuration would be used).
 - Type: **boolean**.
 - Possible values: **false, true**.
 - Default value: **false**.

How does provisioning of the configuration file work?

The provisioning itself is accomplished by obtaining the configuration data from the HTTP(S) server using the URLs specified in the provisioning configuration file and the authentication credentials provided by the user. The HTTP server generates and returns the response in the form of a configuration XML document (please

consult the *Zoiper configuration file documentation* for more information about the structure of that XML document).

By default, Zoiper persists (i.e. stores locally) the provisioned changes to the local configuration (i.e. the provisioned sections). There are exceptions for some sections, which never get persisted in the local configuration if present in the provisioned configuration.

If an `accounts` section is present in the provisioned configuration data and the `delete_accounts_beforehand` option is enabled, this means that only the provisioned accounts will be used and the local ones will get removed from the local configuration. The same behavior is applied to the contact services when the `delete_contact_services_beforehand` option is enabled.

Please do not confuse the *local configuration file* with the *provisioning configuration file*. Their contents are of different XML document types (i.e. they have entirely different structures, and have no options in common): the first one contains the local configuration of Zoiper, and the second one contains the options which determine how provisioning should be performed. Also, keep in mind that provisioning cannot take place whenever there is an active call: it will take place after the moment the call gets hung up.

Error response from the provisioning server

If the server is unable to provide the remote configuration data for some reason, it would respond with an XML-formatted error message with the following structure:

- `error`: the root element of the document. It should contain the error message, which is retrieved and displayed to the user by Zoiper.

Example contents of the provisioning configuration file

```
<?xml version="1.0"?>
<provision>
  <base_urls>
    <base_url>http://yourserver.com/provision/?username=[user]&password=[pass]</base_url>
  </base_urls>
  <options>
    <use_authentication>>false</use_authentication>
    <user_input_authentication>>false</user_input_authentication>
    <authentication_type>url</authentication_type>
    <user></user>
    <password></password>
    <password_format>0</password_format>
    <delete_accounts_beforehand>>true</delete_accounts_beforehand>

    <delete_contact_services_beforehand>>false</delete_contact_services_beforehand>
  </options>
</provision>
```