

# API Usage Examples

This page describes a number of small applications that have been developed to help customers manage their users. These are Windows applications, and use the AdminXML API.

DISCLAIMER: although all of these programs have undergone some testing, they have been developed for specific requirements, so may not be suitable in all scenarios. Read the documentation before using.

## Contents

- 1 Preparation
  - ◆ 1.1 Special Considerations for AdminRequest
- 2 General Advice
- 3 The Swivel Client
- 4 Assign Tokens
- 5 Attribute Manager
- 6 Change PIN
- 7 Bulk Set PINs

## Preparation

In order to use these applications, you must define an Agent in the Sentry administration console (Server -> Agents). The Hostname/IP must correspond to the address of the Windows PC that the applications will be running on. If you need to allow multiple addresses, you can use network masks in the Agent definitions, if appropriate. You can define multiple Agents in certain circumstances, but see the next paragraph.

### Special Considerations for AdminRequest

Where applications add or delete users, the corresponding Agent must be set to Act as a Repository. In this case, the name of the Agent defines the name of the Repository. This has two consequences:

1. You can only define one Agent for each Repository
2. If you need to manage users that are in an existing Repository, the Agent name must exactly match the Repository name

In the latter case, if the Repository is linked to an external source, such as Active Directory, be aware that synchronising with the Repository may undo changes made by the API (and vice versa).

## General Advice

These applications are not provided with installers. Simply unzip all files (in most cases there are 3) into the same folder and run from the command line.

All these applications (with one exception) are written to be run from the command line, and require additional arguments on the command line.

They also come with a configuration file, named as the executable program but with ".config" on the end. Before you use the application for the first time, you will need to edit this file and enter values appropriate for your environment. Below is an example of this file:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="userSettings" type="System.Configuration.UserSettingsGroup, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
    <section name="AssignTokens.Settings" type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  </configSections>
  <userSettings>
    <Properties.Settings>
      <setting name="PINsafeUrl" serializeAs="String">
        <value>https://primary.swiveldev.local:8080/sentry</value>
      </setting>
      <setting name="PINsafeSecret" serializeAs="String">
        <value>secret</value>
      </setting>
      <setting name="AllowSelfSignedSSL" serializeAs="String">
        <value>False</value>
      </setting>
      <setting name="Repository" serializeAs="String">
        <value>AD</value>
      </setting>
    </Properties.Settings>
  </userSettings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
</configuration>
```

You should only change the values within the <value>...</value> elements - do not change anything else.

The above settings exist in the majority of applications, and are described below.

**PINsafeUrl** : the full URL of the Sentry administration URL. Be aware that this must be the core Sentry application - these applications will not work when connected to Sentry SSO or Proxy on port 8443. Do not include the final part of the path: "/AdminXML" - this will be added by the application.

**PINsafeSecret** : the secret as included in the Agent definition

**AllowSelfSignedSSL** : True or False, depending on whether the connection should work if the Sentry certificate is self-signed, has expired, or the URL does not match the hostname in the certificate. "False" will allow the connection to work in all cases, but is less secure.

**Repository** : the name of the repository on which the application is to act. This is only required for applications that use the Helpdesk API: as described above, the Admin API requires that the Repository name matches the Agent name. Instructions for each application will state which type of request it uses. In most cases, you can simply specify "" to indicate that any repository can be used.

Where additional settings are required, or the settings names are different, this will be described under the individual applications.

## The Swivel Client

[Download link.](#)

This is the client DLL that is required by all the applications on this page. A copy of it is provided in each download, so this stand-alone copy is provided for customers who wish to develop their own applications. It is written in C#, as are all the applications, so requires the Microsoft.Net framework to operate.

## Assign Tokens

[Download link.](#)

This application allows you to assign OATH tokens to users, either individually or in bulk. It uses the AdminRequest, so the Agent must be set to Act as Repository.

Usage:

```
AssignTokens <username> <tokenid>
```

Assigns a single token with id <tokenid> to the user <username>

```
AssignTokens <filename>
```

Reads a list of users and tokens from the file <filename>.

Each line in the file should contain a username followed by a token serial ID.

The username and token ID can be separated by spaces, tabs or commas.

Both username and token must already exist in the database.

## Attribute Manager

[Download link](#)

This is a form-based application. It allows you to view and alter certain custom attributes for users - specifically Email address, Phone number, and given and family names. The application uses AdminRequest, so the Agent must be set to Act as Repository. However, it can be used to read attributes from other repositories, but not write, so includes the Repository property.

Note that if the attributes are set to "Synchronised" under the Repository Attributes configuration, they will be overwritten by a user sync.

## Change PIN

[Download link.](#)

This application simply allows you to change the PIN for a single user without knowing the current PIN. It uses HelpdeskRequest, so can be applied to users in any repository.

Usage:

```
ChangePIN username newPIN
```

## Bulk Set PINs

[Download link.](#)

This application allows you to set the PIN for a list of users, either to the same initial PIN, or to individual PINs. It uses HelpdeskRequest, so can be applied to users in any repository.

Usage:

```
InitialPINs <pin> <filename>
```

Where <pin> is the PIN to use for all named users, or "?" to read PIN from the file, and <filename> is the name of a file containing a list of usernames.

If the PINs are in the file, they should be after the username, separated by a Tab character.