



# CONFIGURE PUSH LOGIN WITH OPENOTP

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs.

Copyright (c) 2010-2017 RCDevs SA. All rights reserved.

<http://www.rcdevs.com>

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

## Limited Warranty

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to [info@rcdevs.com](mailto:info@rcdevs.com).

# Configure Push Login with OpenOTP

[PUSH login](#) [Web-Service](#)

## 1. Background

This document describes how to set up Push Login infrastructure, using WebADM, OpenOTP Push Server and optionally WAProxy.

OpenOTP is the RCDevs MFA Service running on top of the RCDevs WebADM platform. OpenOTP itself is composed of several server applications and components that provide secure and reliable authentication of users connecting to applications, online services, intranet, extranet just to name a few. OpenOTP relies on proven technologies and open standards, such as OATH (the initiative for open authentication), HOTP / TOTP / OCRA, Radius, LDAP.

WAProxy is an HTTP(S) reverse proxy for WebADM. While any reverse proxy should be able to fill the role, WAProxy is pre-configured to work securely with all the features of WebADM and thereby is recommended. WAProxy handles basic load balancing, failover and both server and client certificate setup with the least possible amount of configuration effort.

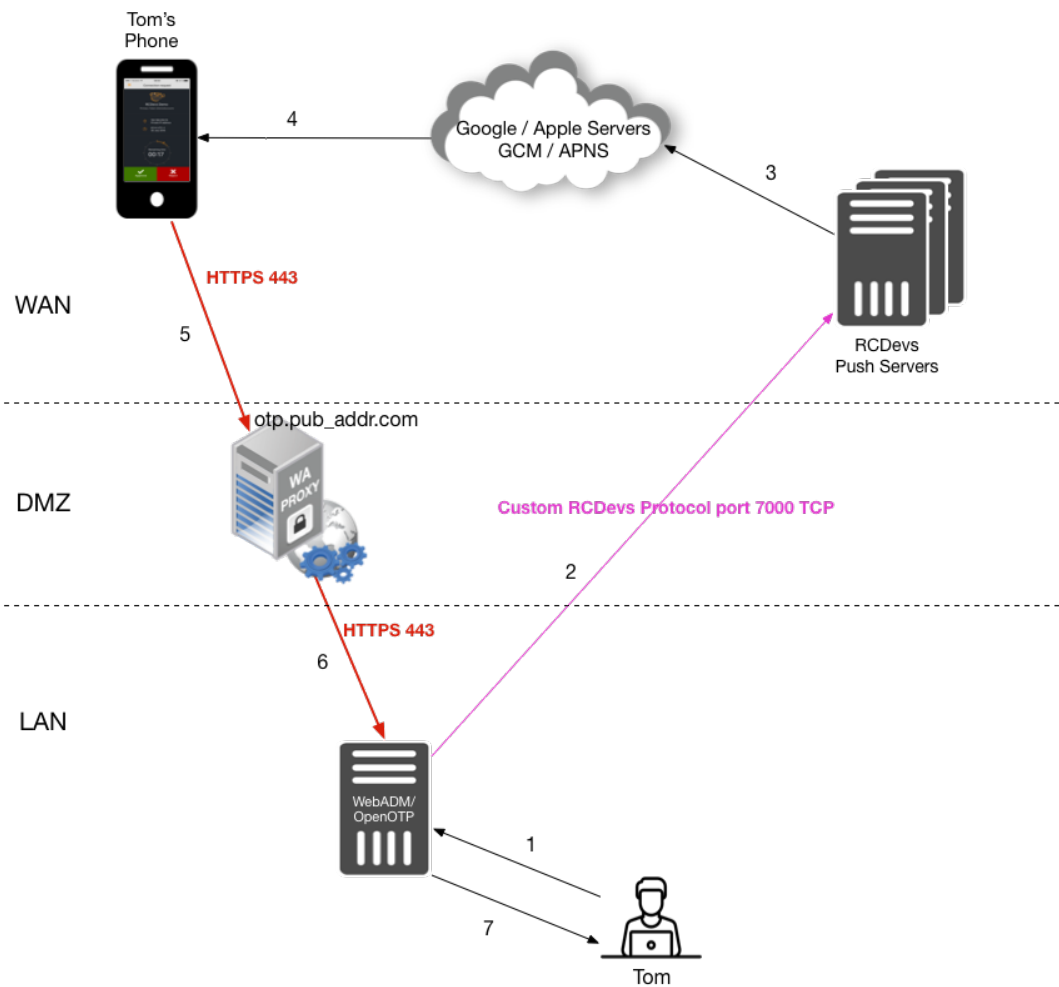
Push Server for OpenOTP Token is a service provided by RCDevs. This service is free for RCDevs customers with an Enterprise Licence. Those using a freeware OpenOTP license (limited to 40 users) must create a free push server account at the following link [RCDevs Push Registration](#).

For more information about RCDevs OpenOTP, please refer to documents at [RCDevs OpenOTP](#).

For more information about RCDevs WAProxy, please refer to documents at [RCDevs WAProxy](#).

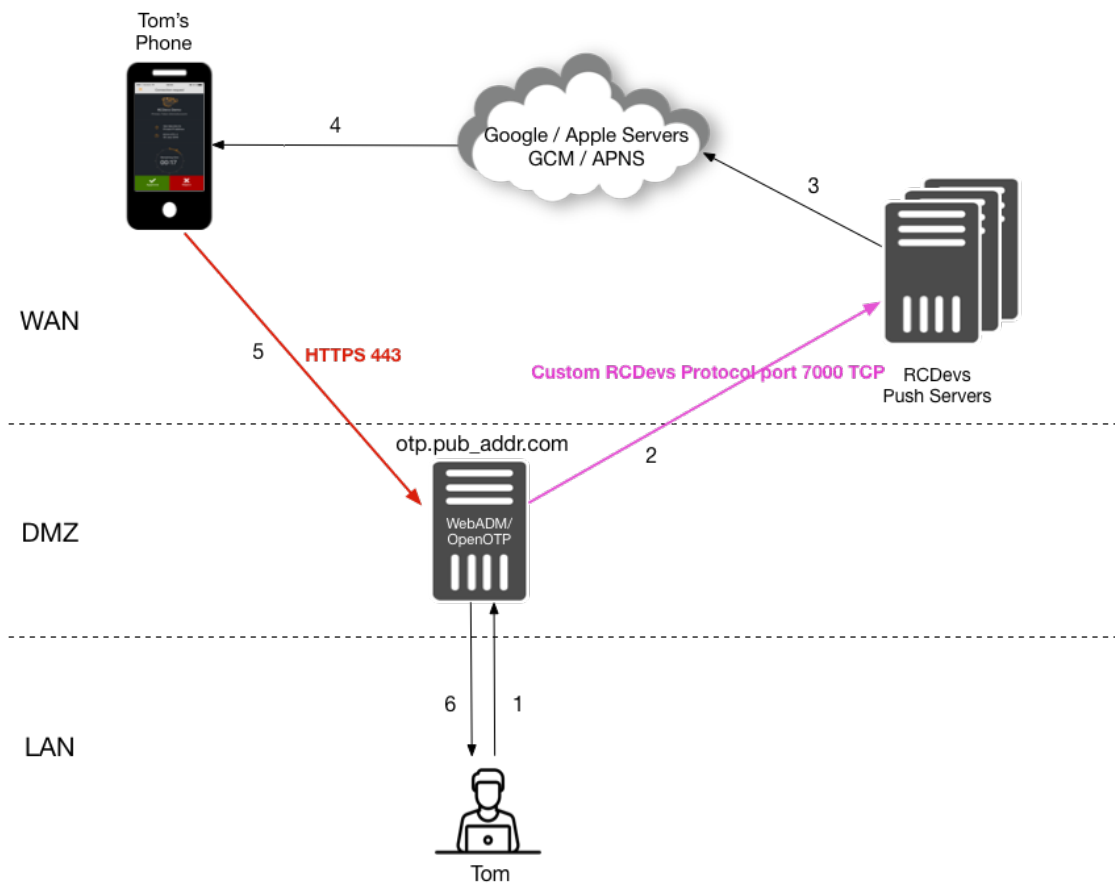
## 2. Deployment Scenarios

### Scenario 1



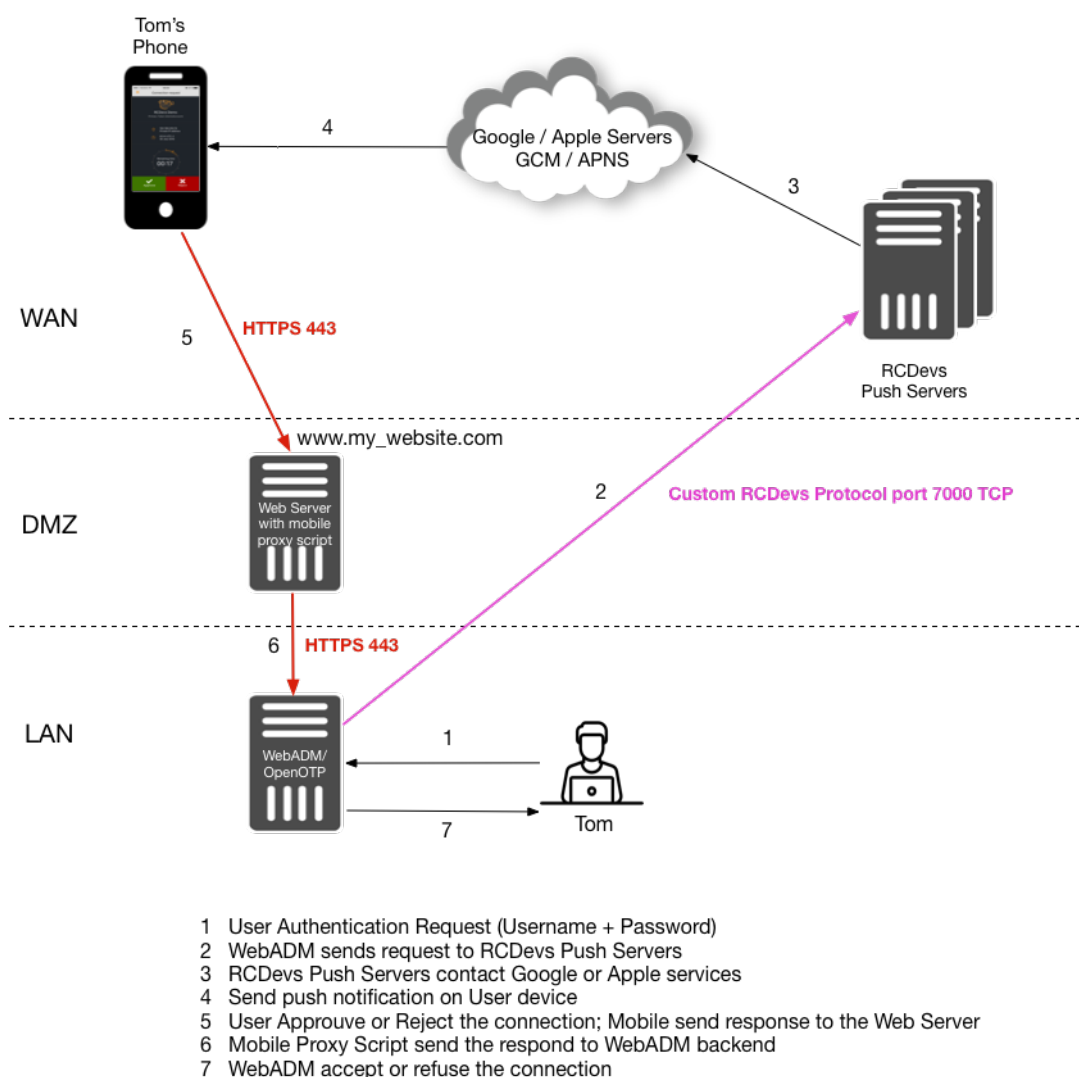
- 1 User Authentication Request (Username + Password)
- 2 WebADM sends request to RCDevs Push Servers
- 3 RCDevs Push Servers contact Google or Apple services
- 4 Send push notification on User device
- 5 User approves or rejects the connection; Mobile sends response to WAProxy
- 6 WAProxy sends the response on WebADM Back-Ends
- 7 WebADM accepts or refuses the connection

## Scenario 2



- 1 User Authentication Request (Username + Password)
- 2 WebADM sends request to RCDevs Push Servers
- 3 RCDevs Push Servers contact Google or Apple services
- 4 Send push notification on User device
- 5 User Approve or Reject the connection; Mobile sends response to WebADM/OpenOTP
- 6 WebADM accepts or refuses the connection

### Scenario 3



### 3. Installation of Required Components

For installation and basic configuration of WebADM, OpenOTP and WAProxy, please refer to manuals available at [RCDevs Documentation](#).

### 4. Deployment Scenarios Described

**Scenario 1:** This scenario includes RCDevs WAProxy on DMZ, avoiding WebADM/OpenOTP servers to be exposed on the Internet. This setup is recommended as being the most secure of the three options.

**Scenario 2:** This is a relatively simple setup with no reverse proxy at all. WebADM/OpenOTP are exposed to the Internet and communicate directly with Push Servers and with end-users' mobiles. This scenario is recommended for testing purposes.

**Scenario 3:** This scenario is similar to Scenario 1, however with the possibility of using an existing Internet-facing Web server as web reverse proxy, rather than WAProxy. You can install RCDevs sample proxy script on your existing web server and this script will act as a reverse proxy as to your WebADM/OpenOTP servers.

The RCDevs sample proxy script can be found on your WebADM installation.

```
/opt/webadm/websrvs/openotp/doc/token_proxy.php
```

#### Note

Please note that your web server have to support PHP and CURL.

## 5. Scenario 1: WebADM Configuration & WAProxy

This chapter describes the configuration of Scenario 1. The chapter assumes you have a working WAProxy already in place. If you do not have WAProxy yet set up, please refer to WAProxy installation and configuration manuals at

<http://www.rcdevs.com/downloads/documents/>.

Once your WAProxy is ready, go to your WebADM server to set up the communication to work between your WebADM and WAProxy.

Edit the file `/opt/webadm/conf/webadm.conf`.

Find the section with WAProxy configuration and edit it:

```
# Private WAProxy IP who will contact the WebADM server
waproxy_proxies "192.168.3.150"

# The public DNS name of your WAProxy server
waproxy_pubaddr "wa.yorcdevs.com"
```

After editing `webadm.conf` file, restart WebADM services:

```
service webadm restart
```

After that, log in on WebADM GUI as Administrator.

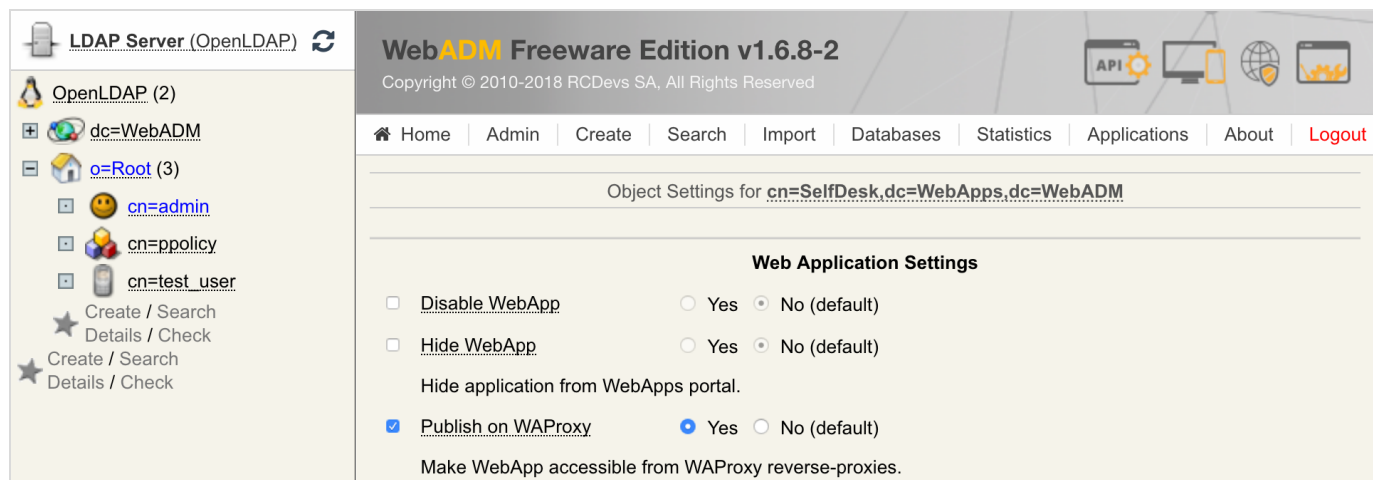
Now, for each application available in WebADM, you should see a new option named Proxy WebSrv which appears in the products configuration pages.

When this option is activated for a given application, the application will then become available and accessible through your

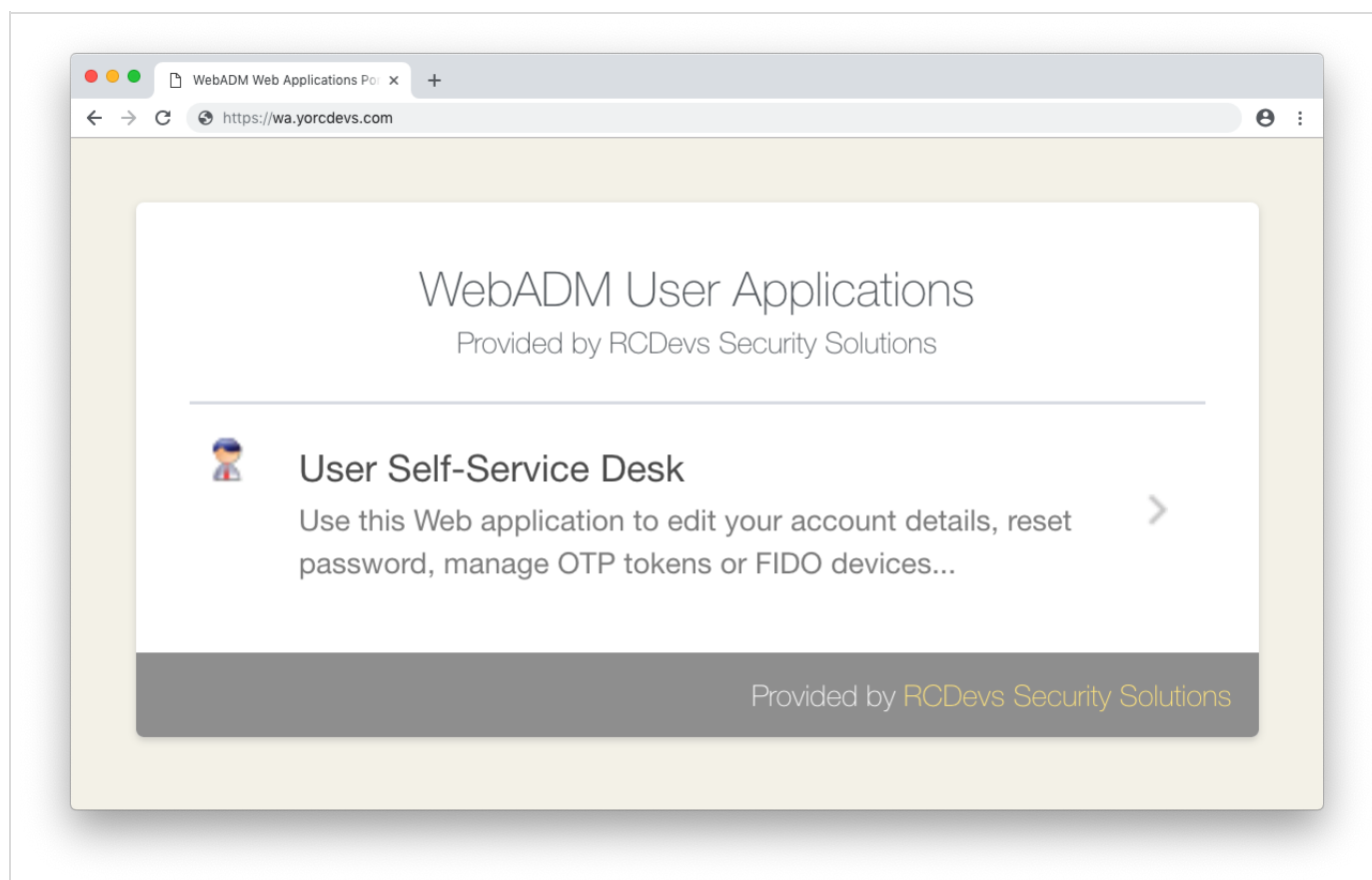
public WAProxy.

For example, to publish WebADM Self-Desk application:

Go to Self-Desk configuration settings and enable WAProxy WebApp as depicted in the picture below.



Save changes. Go to your WAProxy server URL and you should see your Self-Desk there published.



## 6. Scenario 2: WebADM/OpenOTP in DMZ

If your WebADM/OpenOTP servers are already exposed on the internet, or if you simply like to test Push Login, then you do not need to deploy a WAProxy server or the RCDevs Mobile Proxy Script. The URL your phone will contact upon receiving a push notification will simply be the URL of your WebADM/OpenOTP server(s). So there are no configurations needed for this scenario.

However, note that the Mobile URL Endpoint setting in Chapter 8 must be configured as follows:

<https://webadm.yorcdevs.com/ws/openotp/>

Where webadm.yorcdevs.com is a public URL of your WebADM server accessible from the Internet.

## 7. Scenario 3: Mobile Proxy Script

The mobile proxy script must be used when your WebADM/OpenOTP servers are not exposed on the Internet and when WAProxy is not used. This sample PHP script must be placed on a web server already exposed on the Internet.

Configure your Apache, or similar web platform capable of supporting PHP, deploy the proxy script and define the URL that will be called by the mobile in the OpenOTP configuration.

### Example

<https://myserver.com/mobile/>

### Important

You must change the default start pages of your WebServer to “token\_proxy.php”. Or you must simply rename “token\_proxy.php” to “index.php” if “index.php” is one of your default start pages.

Edit the Mobile Proxy Script and configure your WebADM/OpenOTP backends servers:

```
### My 1st WebADM Server
$endpoint_url = "https://192.168.3.54/ws/openotp/"

### My 2nd WebADM Server
$failover_url = "https://192.168.3.55/ws/openotp/"
```

These are the URLs to which your Web server and Mobile Proxy Script will forward the mobile response to, the same manner as WAProxy would.

## 8. OpenOTP Configuration & Push Servers

This chapter describes how to configure WebADM/OpenOTP to connect with RCDevs Push Servers.

Please note that Push server option is only available since WebADM 1.5.3.

To configure OpenOTP with Push Server, you need to go to WebADM web interface and select Application tab, under which you should select the **CONFIGURE** link under **MFA Authentication Server**.

On the following page go to Mobile Push section, where you can enable Mobile Push and configure the Mobile EndPoint URL. This



is the URL contacted by mobile and thereby must be accessible from the internet.

LDAP Server (OpenLDAP)

OpenLDAP (2)

- dc=WebADM
- o=Root (3)
  - cn=admin
  - cn=ppolicy
  - cn=test\_user

Create / Search Details / Check

Create / Search Details / Check

**WebADM Freeware Edition v1.6.8-2**  
Copyright © 2010-2018 RCDevs SA. All Rights Reserved

Home Admin Create Search Import Databases Statistics Applications About Logout

**Mobile Push**

☒ Enable Mobile Push ☒ Yes ☐ No (default)

Enables push-based authentication and online enrolment of RCDevs Authenticator.  
You need to register for a Push service account and configure push servers in servers.xml.  
Note: You don't need a Push service account if you have a RCDevs' Enterprise license.

☒ Mobile EndPoint URL

Public communication URL used by RCDevs mobile Token (with a mandatory trailing slash).  
This setting is not required if you do not use WebADM Publishing Proxy (WAProxy).

☐ Compatibility Mode ☒ Yes (default) ☐ No

Enable Google Authenticator compatibility (allow generic QRCode Tokens and RCDevs Authenticator).  
When disabled, only RCDevs Authenticator with online enrolment and push notifications can be used.

## Note

The /ws/openotp/ at the end of the URL is default for WAProxy (scenario 1).

☒ Mobile Response Timeout

Time to wait for mobile response with Token Simple Push before switching to a fallback method.  
Changing the default value requires to adjust client timeouts (ex. RADIUS request timeout) accordingly!  
Note: This timeout applies to MSS mobile responses as well (MobileID SMS delivery mode).

## ⚠ Mobile Response Timeout

The Mobile Response Timeout setting can be adjusted if you want to have the push notification available more than 30 seconds on your phone. Be careful with this setting in case you increase it, it will impact all the other timeout like the challenge timeout, or idletimeout on other product. If you are not sure how to configure all other timeout on your clients consuming OpenOTP then keep the default value.

e.g : If you configure the Mobile Response Timeout to 60 seconds, you have to increase the login timeout to 75 seconds on every clients/systems using OpenOTP.

If you are setting up scenario 2 then set the address of your WebADM server, i.e:

<https://webadmsrv.mycompany.com/ws/openotp/>.

`/ws/openotp/` is default URL for WebADM/OpenOTP sitting in DMZ.

If you are setting up scenario 3 then set URL to point to where you installed Mobile Proxy Script.

## Example

<https://myWebSrvURL/mobile/>

The last step is to configure the push server location to WebADM. This is done in `/opt/webadm/conf/servers.xml`.

```
<PushServer name="Push Server"
  host="91.134.128.157"
  port="7000"
  user=""
  password=""
  ca_file"" />
```

#### Note

Customers with RCDevs Enterprise Licence will not need to set user & password. Those with a freeware license must register for a push server account at [RCDevs Push Registration](#) to obtain the credentials. The activation of a newly-created push server account can take up to one hour.

## 9. OpenOTP Token Enrolment with User Self-Services

At this point your configuration is complete and you are ready to start enrolling OpenOTP Token. This chapter describes enrolling OpenOTP Token from the User Self-Service Desk application, available on your WebADM server.

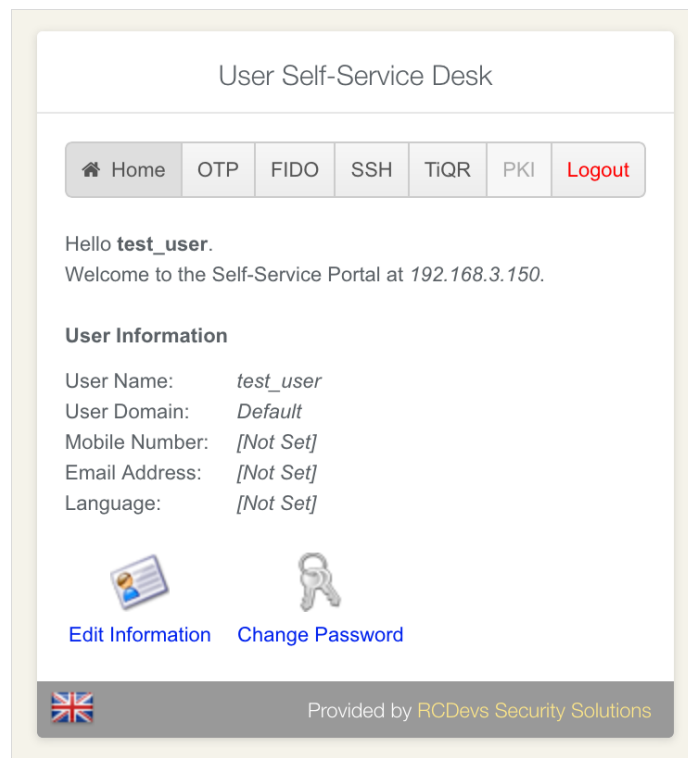
OpenOTP Token is currently available on App Store for iPhone and on Google Play for Android.

Here we describe the User Self-Service Desk accessed through a WAProxy (Scenario 1). The methodology is, however, the same for all three scenarios.

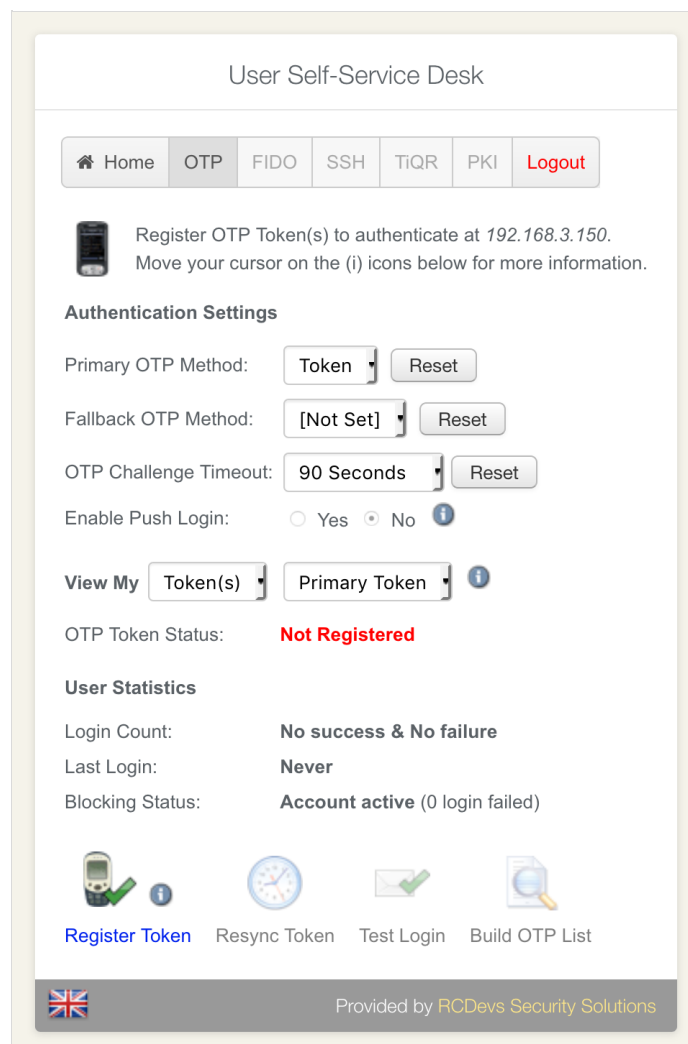
#### Note

Also, note that user enrolment can be run directly through the WebADM administrator interface.

Logon in to User Self-Service Desk:



Once inside, click on OTP application menu and Register Token:




Choose the option QRCode-based. You can then choose between time-based or event-based.

User Self-Service Desk

You must first register your Software or Hardware Token to start using it.  
The registration consists in synchronizing a Secret Key and an initial Token state.

Instructions to register a QRCode-based Software Token:

1. [Install the Software Token](#) on your mobile device.
2. Start your software Token and Scan the QRCode displayed below.
3. Click the 'Register' button below after scanning.



☐ I use a Hardware Token (Inventoried)

☐ I use a Yubikey Token (Inventoried / YubiCloud)

☒ I use a QRCode-based Authenticator (Time-based)

☐ I use a QRCode-based Authenticator (Event-based)


☐ I use another Token (Manual Registration) ⓘ

Register As: 

Primary Token

QRCode:

[\(Enlarge\)](#)




ⓘ

Enter OTP:

Register

Cancel



Provided by [RCDevs Security Solutions](#)

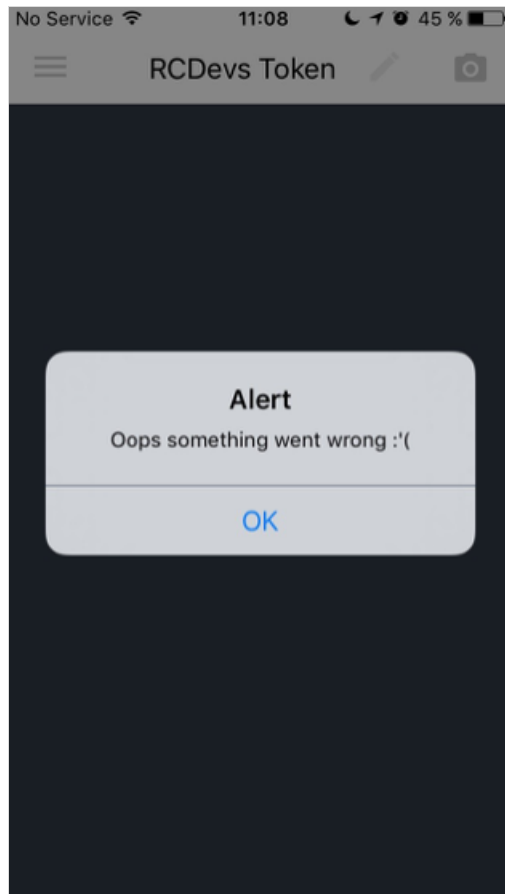
Always prefer time-based if you are unsure which to choose. Open the OpenOTP Token mobile application, scan the QRCode.

### ⚠ Important

You must be connected to the Internet with your smartphone to enroll your OpenOTP Token.

### ⚠ Alert

If you end up with a screen like below, it's probably because your smartphone cannot communicate with the configured Mobile End Point URL (configured in chapter 8). Check both your mobile internet connectivity and Mobile End Point URL.



Otherwise, your OpenOTP Token will be automatically enrolled and ready for use! You should also see your Token enrolled within the User Self-Service Desk.

User Self-Service Desk

---

Home
OTP
FIDO
SSH
TIQR
PKI
Logout

Register OTP Token(s) to authenticate at 192.168.3.150.  
Move your cursor on the (i) icons below for more information.

**Authentication Settings**

Primary OTP Method: Token Reset

Fallback OTP Method: Disabled Reset

OTP Challenge Timeout: 90 Seconds Reset

Enable Push Login: ☐ Yes ☒ No (i)

View My Token(s) Primary Token (i)

OTP Token Status: Ok (Disable) (Unregister) (i)

Token Type: **OATH Time-based** (160 bits)

Time Interval: **30 Seconds**

Max Time Offset: **120 Seconds**

**User Statistics**

Login Count: **1 success & 1 failure**

Last Login: **2018-11-23 15:48:18**

Blocking Status: **Account active** (0 login failed)

Register Token
Resync Token
Test Login
Build OTP List

Provided by RCDevs Security Solutions

## 10. Enforcing Simple-Push Login & Testing

You can enforce simple-push login (one-tap access with approve/reject) for one single user, for a group of users, for all your users, or for a given application/device (see WebADM Client Policies). In this example, we enable simple-push login for a one individual user.

Go to WebADM GUI interface and click on the user that you like to configure for push notification. In object details, click on CONFIGURE WebADM Settings, then choose the OpenOTP Application.

The screenshot shows the WebADM Freeware Edition v1.6.8-2 interface. On the left is a sidebar for LDAP Server (OpenLDAP) with a tree view containing: OpenLDAP (2), dc=WebADM, o=Root (3), cn=admin, cn=ppolicy, and cn=test\_user. The main content area displays details for the object cn=test\_user.o=Root. It is divided into three sections: LDAP Actions, Object Details, and Application Actions.

LDAP Actions	Object Details	Application Actions
<a href="#">Delete this object</a> <a href="#">Copy this object</a> <a href="#">Move this object</a> <a href="#">Export to LDIF</a> <a href="#">Change password</a> <a href="#">Create certificate</a> <a href="#">Unlock WebApp access</a> <a href="#">Advanced edit mode</a>	Object class(es): webadmAccount, person Account is unique: <b>Yes</b> (in o=root) WebADM settings: <b>1 settings [CONFIGURE]</b> WebADM data: <b>9 data [EDIT]</b> User activated: <b>Yes Deactivate</b> Logs and inventory: <a href="#">WebApp</a> , <a href="#">WebSrv</a> , <a href="#">Inventory</a>	<a href="#">Secure Password Reset (1 actions)</a> <a href="#">User Self-Registration (1 actions)</a> <a href="#">MFA Authentication Server (13 actions)</a> <a href="#">SMS Hub Server (1 actions)</a> <a href="#">SSH Public Key Server (3 actions)</a> <a href="#">QR Login &amp; Signing Server (8 actions)</a>

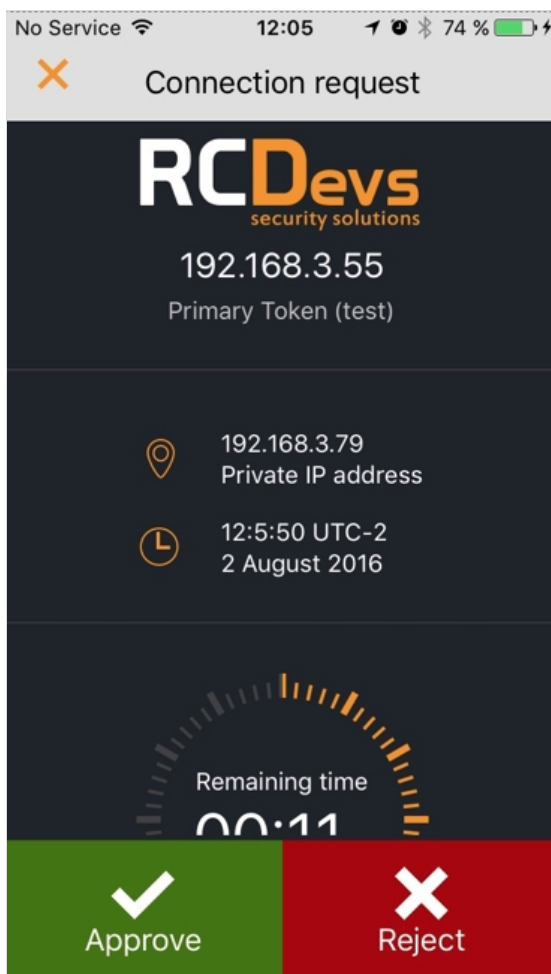
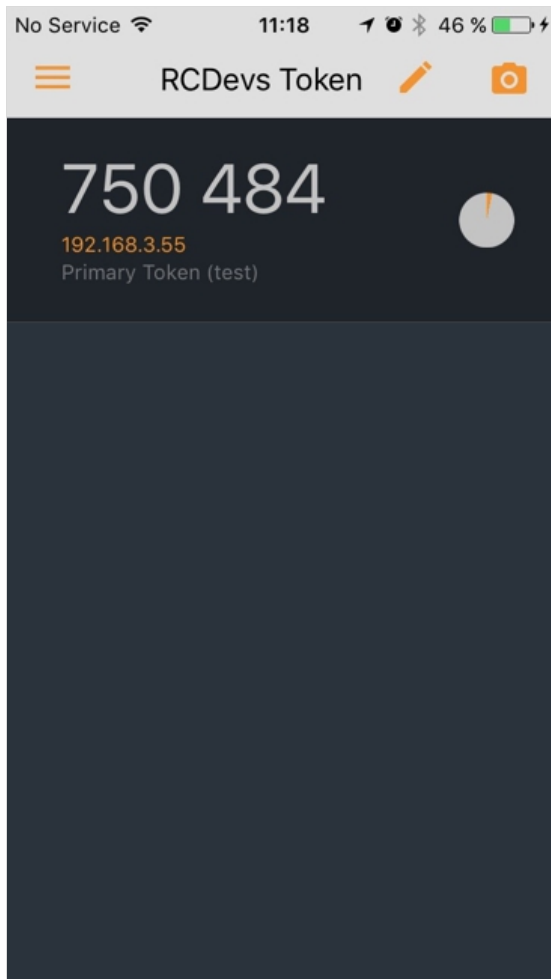
Find the Token Features section (like in the picture below). Enable Simple-Push Login.

The screenshot shows the 'Token Features' section in the WebADM Freeware Edition v1.6.8-2 interface. The sidebar on the left is identical to the previous screenshot. The main content area is titled 'Token Features' and contains two settings:

- ☒ **Simple-Push Login** ☒ Yes ☐ No (default)  
 Enable simplified mobile Push authentication with one-tap 'Approve/Deny'.  
 Simple-Push does not work as fallback and requires Mobile Push activation with RCDevs Authenticator.
- ☐ **Token Expiration Time** 0  
 Time after which a Software Token expires and must be re-enrolled (in days).  
 Set '0' to disable the expiration on newly registered software Tokens.

Save changes and go back to the User Self-Service Desk to test the push login.

In User Self-Service, choose Test Login. A push notification should appear in your Smartphone, similar to picture below:





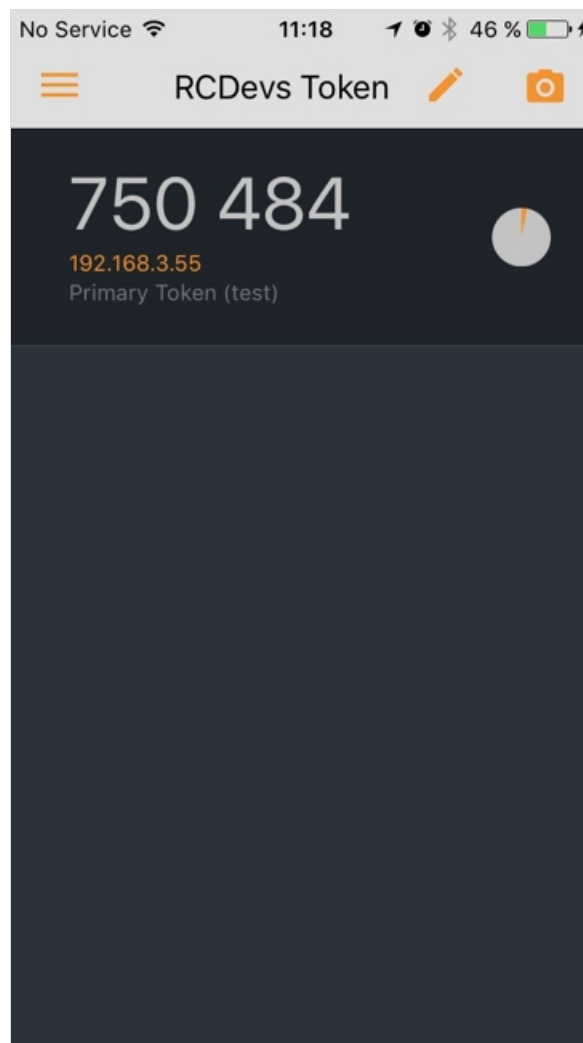
Click on Approve and authentication should complete successfully!

## 11. OTP Types and NoWaitState flag

Receiving a push message requires the mobile to have a network connection. In case the mobile for some reason cannot receive the push message, OpenOTP implements a mechanism to bypass the wait for the simple push timeout.

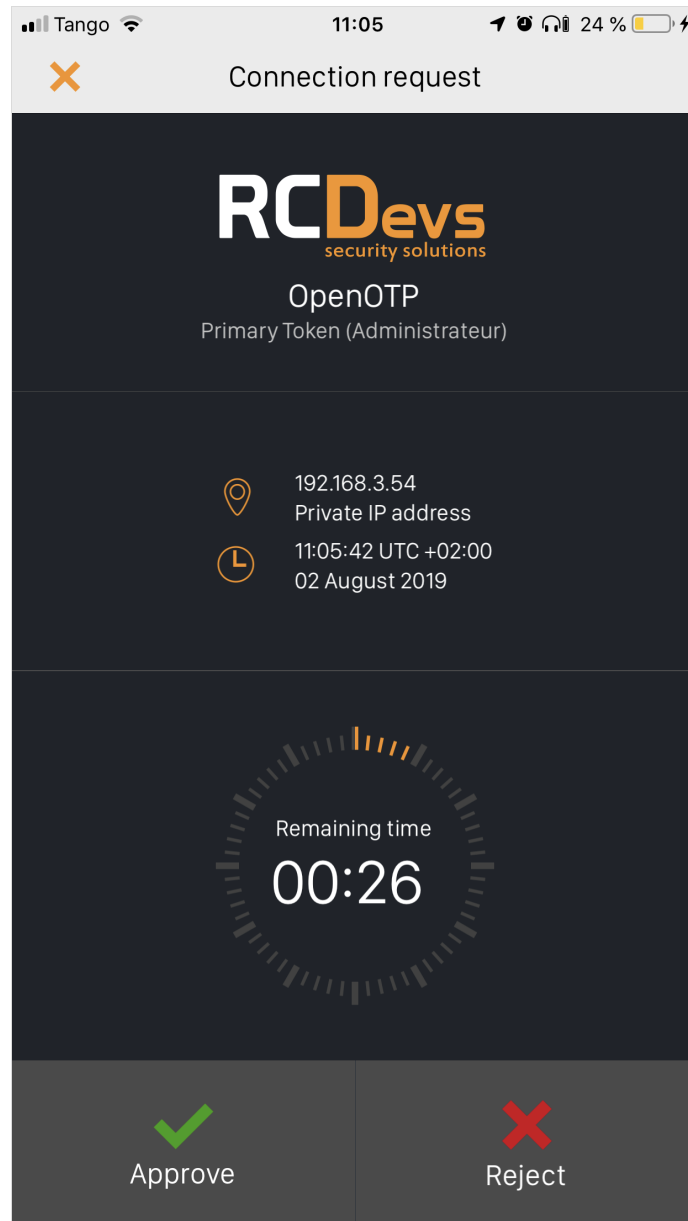
### 11.1 Normal Time or Event based OTPs

The regular OTPs displayed when your OpenOTP Token application is open are based on time or event counter. When an authentication is attempted with simple approve/reject workflow the OpenOTP server waits for a configured time for a response from a mobile. If there is no response, the user is asked for an OTP. If they provide the regular displayed OTP, then the `NoWaitState` flag is added to the user account. When that flag is present on users account, the simple push approve/reject is temporary disabled and for the next user login, the user will receive a displayed OTP push. If the user authenticates with the displayed OTP push, then the `NoWaitState` flag is removed and the simple push is re-enabled.



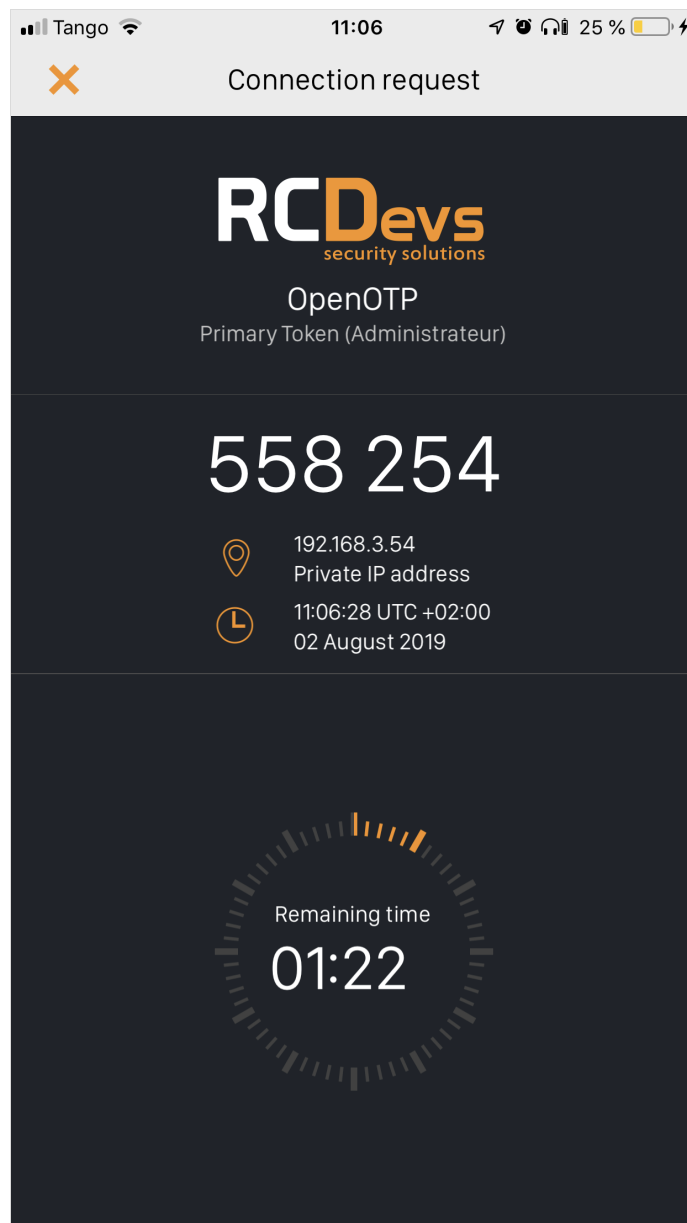
### 11.2 Simple Push (Approve/Reject)

This is a simple push. This kind of push is received when the `NoWaitState` flag is NOT set on the user account.



### 11.3 Displayed push

The displayed push is sent when the `NoWaitState` flag is present on the user account. That allow OpenOTP server to send a challenge request to the client system to allow the user to enter the 6 digits OTP without waiting for the simple push timeout.



If a user successfully authenticates with displayed push OTP, it means the user's phone is receiving push notifications and the `NoWaitState` flag can be removed from user's account, re-enabling simple approve/reject push for that user.

## 12. Video Demonstration



Play Video on Youtube

*This manual was prepared with great care. However, RCDevs S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs S.A. The latter especially applies for data processing systems. RCDevs S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs S.A. © 2019 RCDevs SA, All Rights Reserved*