



Manual

DoubleClue Developer's Guide (Desktop)

Version: 2.1.0

Contents

1.	Introduction.....	3
1.1	Disclaimer.....	3
2.1	App-GUI	3
2.2	SDK-Library.....	3
2.3	Requirements.....	4
2.4	Contents of Deliverables.....	4
3.	SdkConfig.dcem	4
4.	Getting Started with “LibDoubleClue”	5
4.1	Dependency Libraries	5
4.2	Initialization Steps	Fehler! Textmarke nicht definiert.

1. Introduction

1.1 Disclaimer

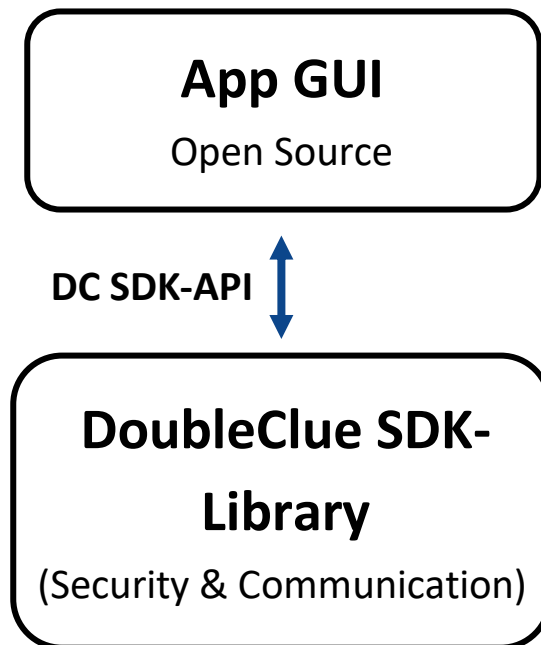
This guide is intended for developers who already possess some knowledge of JavaFx Application. The Desktop Application is written with JavaFX and runs on Windows or MacOS.

With DoubleClue Desktop App, the user will be able to register at and login to DoubleClue Enterprise Management (DCEM). After login the user can receive messages and send back signed message responses.

The DoubleClue Desktop Application consists of two main parts:

- App-GUI
- DoubleClue SoftwareDevelopmentKit (SDK) Library

A complete Desktop Application including installer is available as well.



1.2 App-GUI

The Application-GUI is part implements all user interfaces and GUI navigations. This is a paradigm of a typical DoubleClue App which is delivered in source files. You are free to change the sources according to your needs after signing the DoubleClue License agreement.

1.3 SDK-Library

The SDK-Library is responsible for all cryptography and communication security with DCEM.

1.4 Requirements

- JavaFx SDK
- Eclipse IDE or IntelliJ IDEA

1.5 Contents of Deliverables

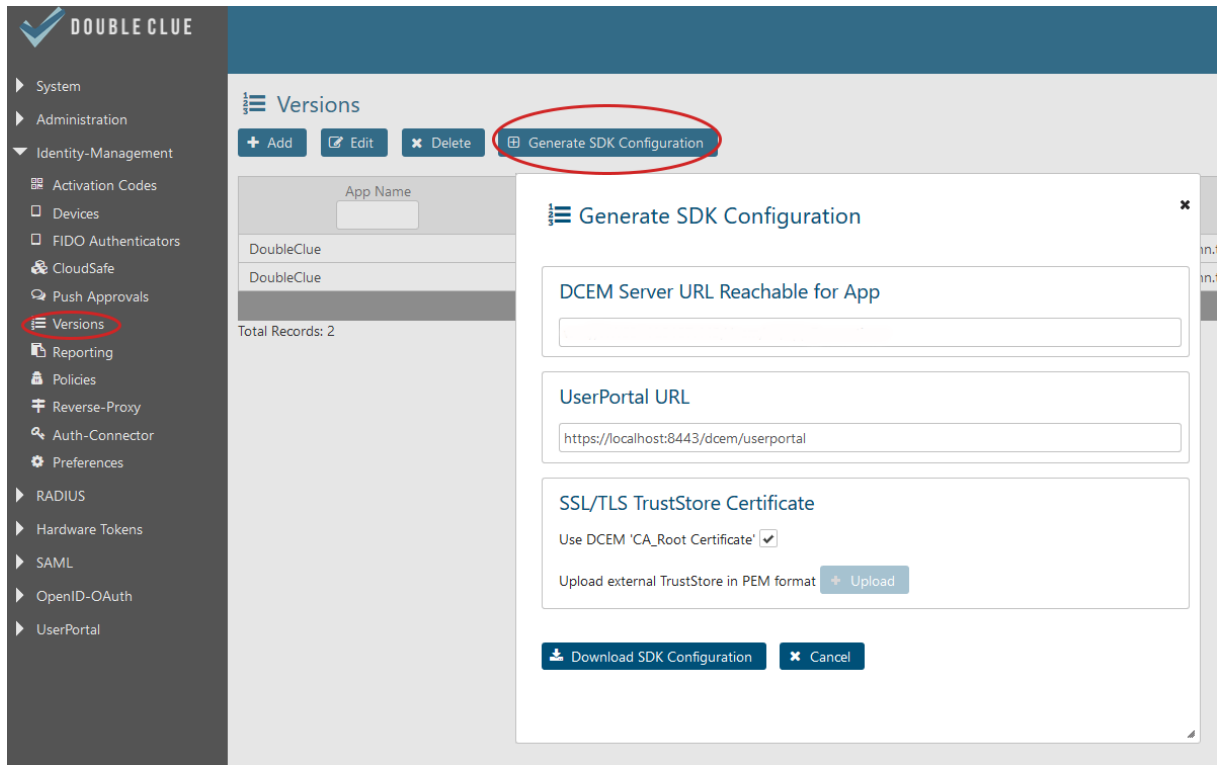
- **doc**
This directory contains several documentation.
- **lib**
The directory contains all the dependent libraries you need to run the DoubleClue Desktop Application.
- **source**
This directory contains the source files of the GUI packed in a jar file.
- **windows**
Here you will find the windows executables


2. SdkConfig.dcem

In order to create a secure connection to DCEM, you will to download the “**SdkConfig.dcem**” from DCEM (Menu Identity-Management>Versions). This file has to be copied into the folder “**src\main\java\resources\config**” directory.

This file contains several secure information which are required to establish a trust connection to your DCEM Server: the Server URL, the public-Key of the trust DCEM and other secure information.

This files contains also the CA-Trust-Certificates for the SSL/TLS connection.



 The file is signed and any changes made in it will corrupt the file contents.

Note: DoubleClue uses an extra X.509 trust certificate infrastructure between SDK-Clients and DCEM, which is independent from the SSL/TLS certificates.

The file has to be downloaded from DCEM (main menu “App-Management”, sub menu “Versions”, button “Generate SDK-Configuration”).

For more information please have a look at the DCEM Manual (“[DCEM_Manual_EN.pdf](#)”) or ask your DCEM administrator.

3. Getting Started with “LibDoubleClue”

3.1 Dependency Libraries

Add all the jar files in the subdirector „lib“ to your project library dependencies.

- Create the class `AppSdkListnerImpl` which implements the Interface **`AppSdkListener.java`**
- Create an instance of the `AppSdkImpl.class`
- Create an instance of the `AppSdkListnerImpl.class`
- Create instance of `AsVersion.class`
- Read the contents of `SdkConfig.decm` from the resources.
- Create the `Map<String Object>` properties

- g) Add the Locale to the properties using the key `AsConstants.PROPERTY_LANGUAGE`
- h) Optionally you can add the Sdk-Home Directory by using the key `AsConstants.PROPERTY_HOME_DIRECTORY`
- i) Call the `appSdkImpl.initialize(.....)`;
- j) Call `appSdkImpl.getActivatedUsers()`; to get the already registered users. If no users are registered show the activation view.

JAVA Example:

```
AppSdkImpl appSdkImpl = AppSdkImpl.getInstance();
AppSdkListnerImpl appSdkListnerImpl = new AppSdkListnerImpl(this);
AsVersion asVersion = new AsVersion("AppName", 1, 0,0);

byte[] sdkConfigDcem =
Utils.readInputStream(this.getClass().getResourceAsStream(AppConstants.RESOURCE_SDK_CONF
IG));

Map<String, Object> properties = new HashMap<>();
properties.put(AsConstants.PROPERTY_LANGUAGE, appSettings.getLanguage());

appSdkImpl.initialize(asVersion, appSdkListnerImpl, properties, sdkConfigDcem);
List<String> users = appSdkImpl.getActivatedUsers();
```