

**Manual**

**DoubleClue Developer’s Guide (Desktop)**

**Version: 2.1.0**

**Contents**

[1. Introduction 3](#_Toc25231825)

[1.1 Disclaimer 3](#_Toc25231826)

[2.1 App-GUI 3](#_Toc25231827)

[2.2 SDK-Library 3](#_Toc25231828)

[2.3 Requirements 4](#_Toc25231829)

[2.4 Contents of Deliverables 4](#_Toc25231830)

[3. SdkConfig.dcem 4](#_Toc25231831)

[4. Getting Started with “LibDoubleClue” 5](#_Toc25231832)

[4.1 Depencency Libraries 5](#_Toc25231833)

[4.2 Initialization Steps **Fehler! Textmarke nicht definiert.**](#_Toc25231834)

# Introduction

## 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.

**App GUI**

Open Source

**DoubleClue SDK-Library**

(Security & Communication)

**DC SDK-API**

## 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.

## SDK-Library

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

## Requirements

* JavaFx SDK
* Eclipse IDE or IntelliJ IDEA

## Contents of Deliverables

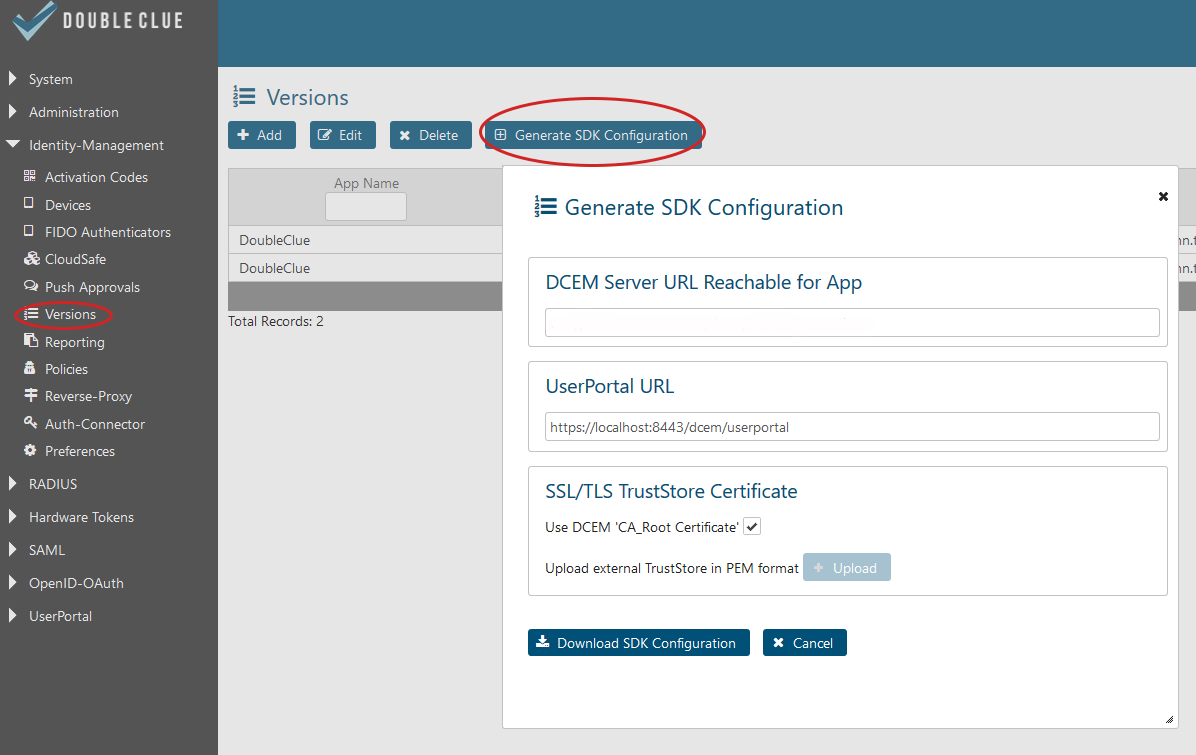
* **doc**   
  This directory contains several documentation.
* **lib**   
  The directory contains all the dependent libraries you need to run the DoubleClue Desktop Application.
* **source**This directory contains the source files of the GUI packed in a jar file.
* **windows**Here you will find the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

# 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.



C:\Users\maike.behnsen\Desktop\Grafiken\caution-sign.png 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.

# Getting Started with “LibDoubleClue”

## Dependency Libraries

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

## Initialization Steps

1. Create the class AppSdkListnerImpl which implements the Interface **AppSdkListener.java**
2. Create an instance of the AppSdkImpl.class
3. Create an instance of the AppSdkListnerImpl.class
4. Create instance of AsVersion.class
5. Read the contents of SdkConfig.decm from the resources.
6. Create the Map<String Object> properties
7. Add the Locale to the properties using the key AsConstants.PROPERTY\_LANGUAGE
8. Optionally you can add the Sdk-Home Directory by using the key AsConstants.PROPERTY\_HOME\_DIRECTORY
9. Call the appSdkImpl.initialize(……);
10. 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\_CONFIG));

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

appSdkImpl.initialize(asVersion, appSdkListnerImpl, properties, sdkConfigDcem);

List<String> users = appSdkImpl.getActivatedUsers();