Security Classification:

EIDA ID Card Toolkit v2.7

Installation and Configuration Guide

Document Details

Organization	Emirates Identity Authority (EIDA)
Document Title	Document Name
Date	31-10-2012
Doc Name / Ref	Toolkit Installation and configuration
Classification	○ Public ○ Internal ◉ Confidential ○ Highly Confidential
Document Type	○ Policy ○ Procedure ○ Form/Template ○ Report ◉ Other

Document History

Date	Version	Author	Comments
25/7/2010	0.1	Ahmad Ibrahim	Draft
15/8/2010	0.2	Hakkim Sebssi	
19/8/2010	0.9	Shiva Murugesan	
19/8/2010	1.0	Malek Bechlaghem	Final Version issued for Toolkit V1.
11/11/2010	1.1	Ahmad Ibrahim	Updated version for Toolkit V.2
3/10/2011	1.2	Ahmad Ibrahim	Updated version for Toolkit V.2.2.2
17/11/2011	v2.2.2 v0.90	Mohammad Abd El Shaheed	Version issued for Toolkit V.2.2.2
30/10/2012	V2.7	Ahmad Ibrahim	Version issued for Toolkit V.2.7.0

Contents

Table of Figures

1 Introduction

This document contains Installation guide for the EIDA ID card toolkit where it describes the software requirements and installation steps of the toolkit; moreover the document also explains the configurations of the different components of the toolkit.

The document is organised as below:

- Section 2 : Provides an overview of the Toolkit components
- Section 3 : Provides an overview of the toolkit installation, requirements and supported operating systems
- Section 4: Describes in details the toolkit installation steps
- Section 5: Describes the configuration details

2 Toolkit components

EIDA ID Card Toolkit SDK contains the below components.

2.1 Core library

This library is base of EIDA toolkit SDK where all the core functions are implemented. It consists of the main native Toolkit library *UAE_IDCardLib.dll* and helper core libraries.

2.2 Java API

This component wraps the core functions of the Toolkit into a simple, object-oriented and high level Java API library and provides easy access to EIDA ID card functions for Java developers.

2.3 C# API

This provides the same level of Java API functionality, but based on Microsoft .Net environment in order to suite .NET developers.

2.4 Web components

The web components are acting as web browsers plug-ins and provide interfaces suitable for script languages used in web pages. It is working as middle layer between the web pages that reads the card public data and the low level toolkit libraries.

For wide interoperability, Toolkit provides two kinds of plug-ins; ActiveX (In general, preferable for Microsoft platform) and Applet (for java platforms).

The standard Toolkit web components heavily depend on toolkit core libraries and API's. The zero footprint web components do not have dependencies on the Toolkit kernel or any other Toolkit components. However they provide access to a limited set of Toolkit functions including Read public data and PKI functions.

2.5 Secure messaging web service

This web service performs remote secure messaging functions in case if EIDA secure messaging module is accessed remotely.

2.6 Samples

- Desktop applications implemented in both Java and .Net demonstrates the Java and .Net API functions.
- HTML page using ActiveX demonstrates the usage of Public Data ActiveX and Digital Signature ActiveX.
- HTML page demonstrates the usage of IDCard Applet and Toolkit API.

- Sample web applications implemented in both Java and .Net demonstrates the usage of the zero footprint web components including the usage of Public Data Parser, Toolkit signature and certificate validation API.

3 System Requirements

3.1 Software Requirements

.Net Framework 3.5: required only if the user of the SDK is using .Net as a development environment, using EIDA .Net API, or using Public Data ActiveX.

Java Runtime Environment 1.6: required only if the user of the SDK is using Java as a development environment, using EIDA Java API, or using Public Data Applet.

Internet Explorer 6 or higher: required if the user is using any of the Public Data ActiveX or Applet.

MorphoSmart USB driver: required if the SAGEM MSO sensor will be used.

3.2 Hardware Requirements

Your computer must meet or exceed the following platform and operating system requirements:

- 2GB RAM
- 100 MB free disk space
- PC\SC complaint Card reader

3.3 Operating Systems

The current version of the Toolkit is designed to work on the below Operating Systems / programming languages.

- Windows XP
- Windows Vista
- Windows 2003 Server
- Windows 2008 Server
- Windows 7

4 Installing EIDA Toolkit SDK

4.1 Installation Prerequisites

Install the following prerequisites before starting the Toolkit installation if you have teh the setup files already which will save time else Toolkit will download it from the web.

- Java Runtime Environment 1.6 (or JRE 6)
- Microsoft .Net Framework v3.5

4.2 Installation Steps

EIDA toolkit SDK is packaged in an easy to use setup wizard, which is self described and well guided. Follow the instructions as below.

- Run EIDA_SDKSetup.exe
- If a previous version or the same version of the toolkit SDK was already found, the following screen will appear asking to remove the previous version of the toolkit

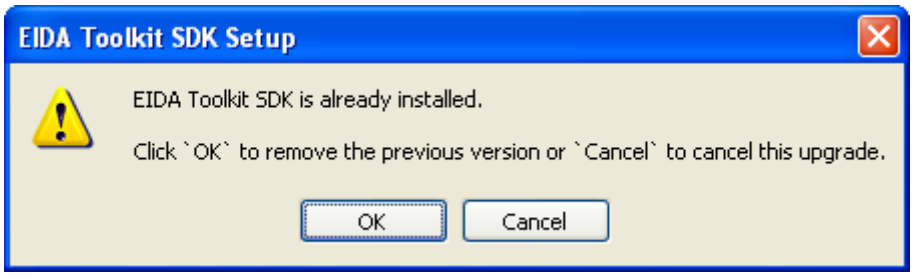


Figure 1 Uninstall Confirmation

- If you click “OK”, an uninstall dialog will appear to remove the previous toolkit version

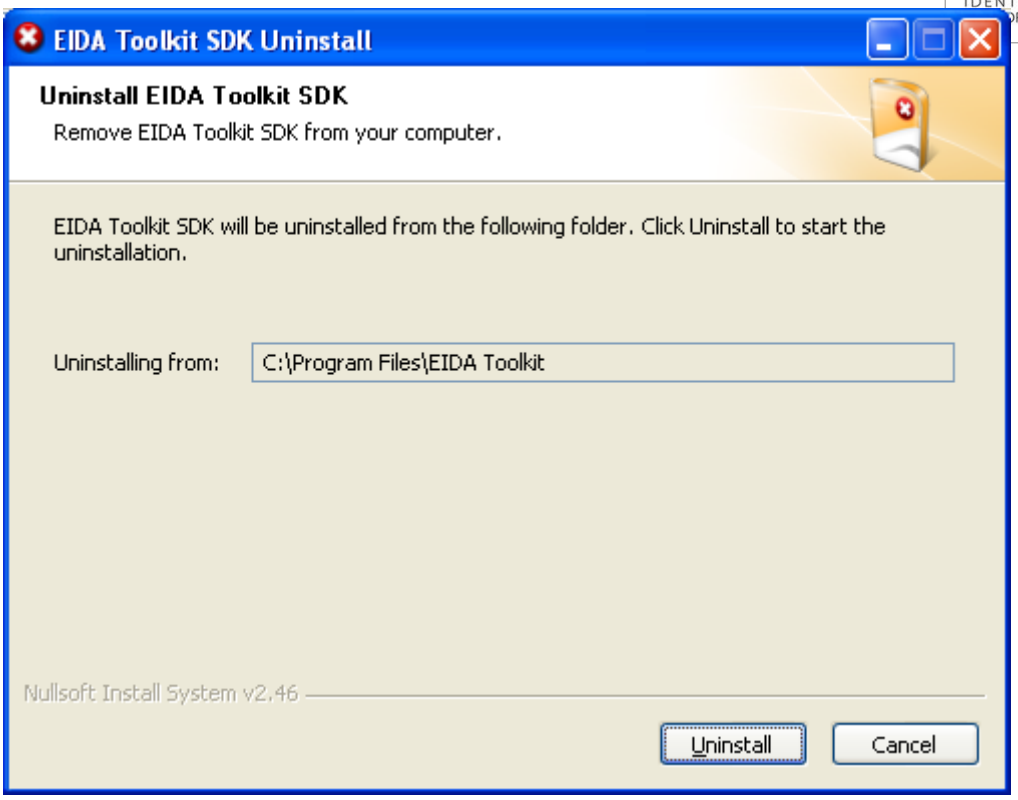


Figure 2 Start uninstall

- Click “Uninstall”, the installer will automatically remove all toolkit components from your system

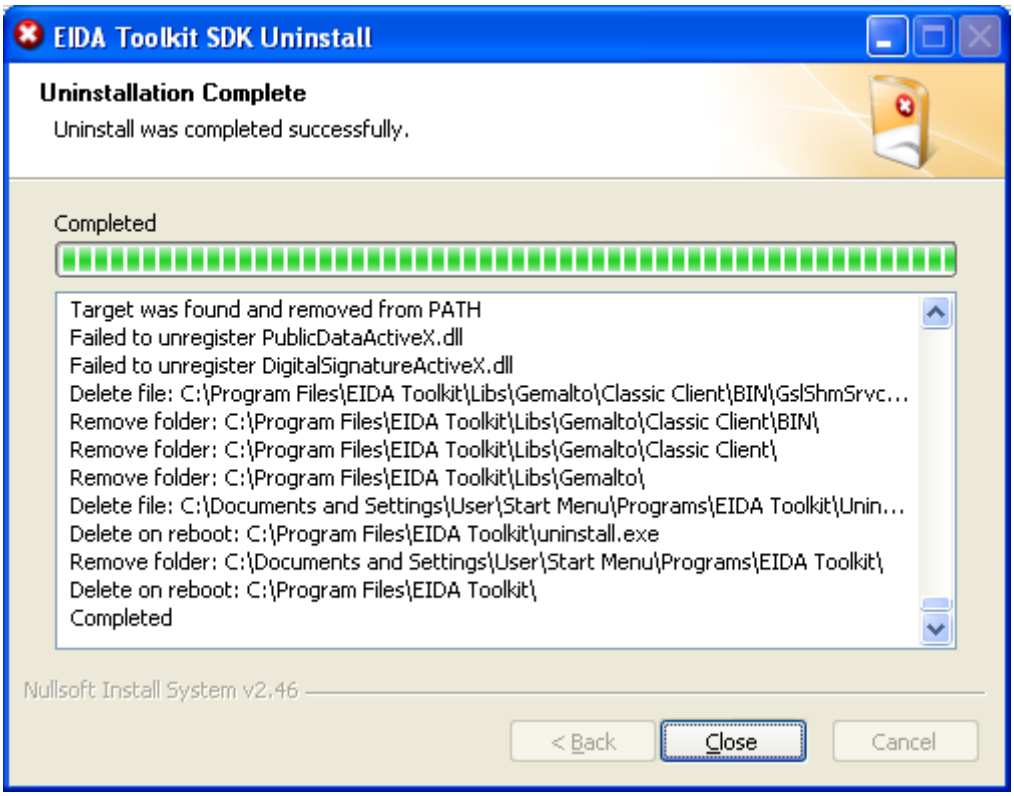


Figure 3 Uninstallation Complete

- Click “Close” to start Toolkit SDK installation wizard
- the setup wizard starts showing the welcome(**Figure 4**)

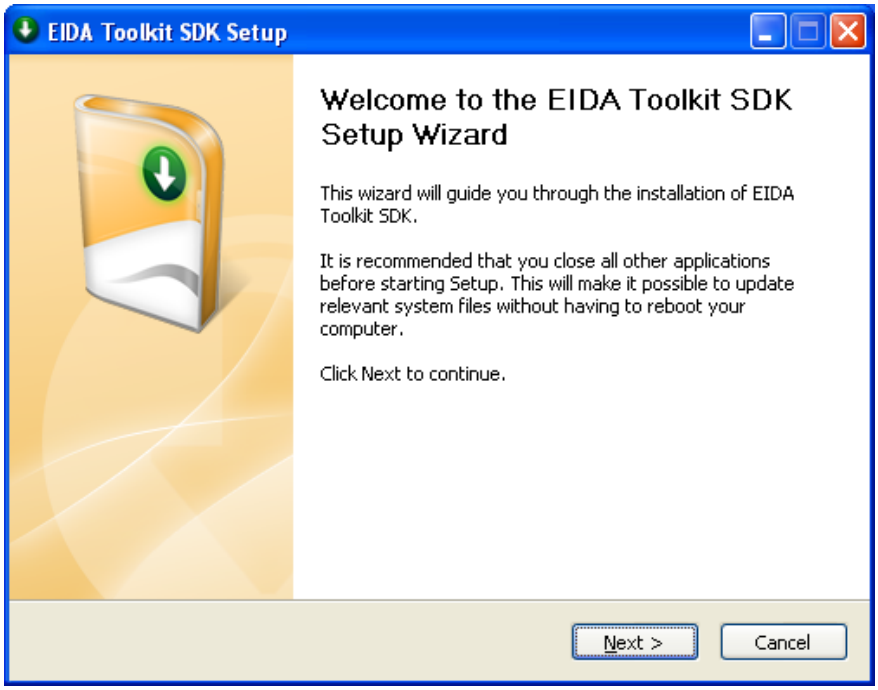


Figure 4: Setup welcome screen

- Click “Next” to see a screen which lets you choose the components to install (
- Figure **5: Component selection**
-).

- Components are described in section 2 of this document.

- Select the components you want to install and click “Next”.

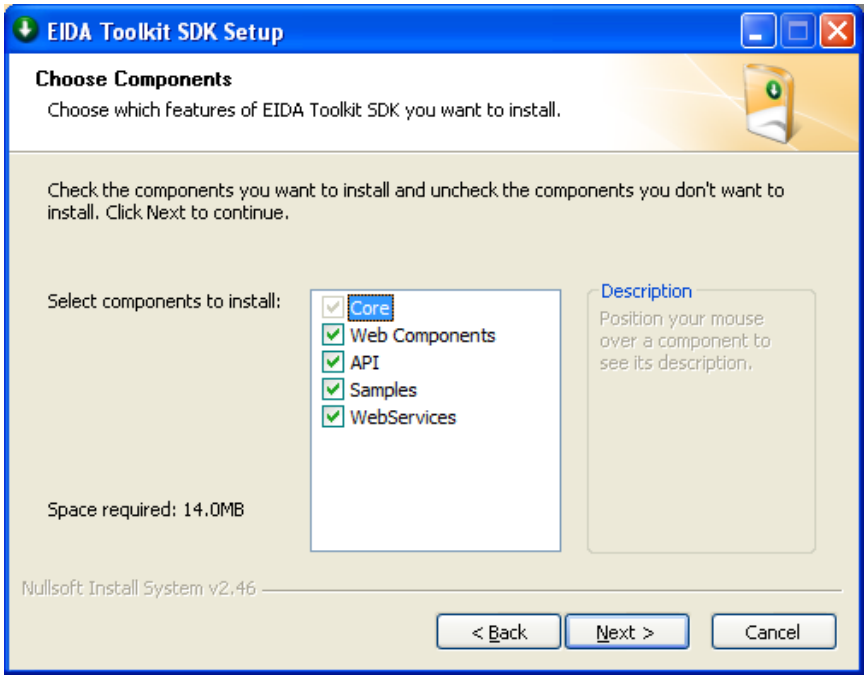


Figure 5: Component selection

- Setup wizard then requests a destination folder for the components to be installed.
- Choose the desired location and the click “Install”.

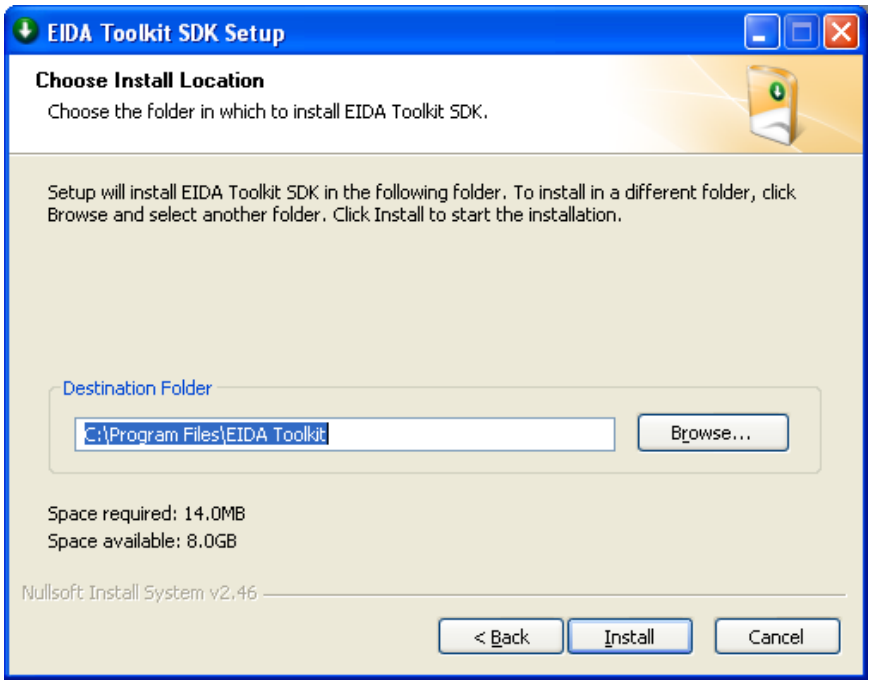


Figure 6: Choose Install Location

- The setup wizard then checks if the Java Runtime 1.6 installed, if it’s not installed the wizard will start downloading it.

- If the Java Runtime 1.6 is already installed, the setup wizard will skip this step, cancel the download and continue installation to install it later.

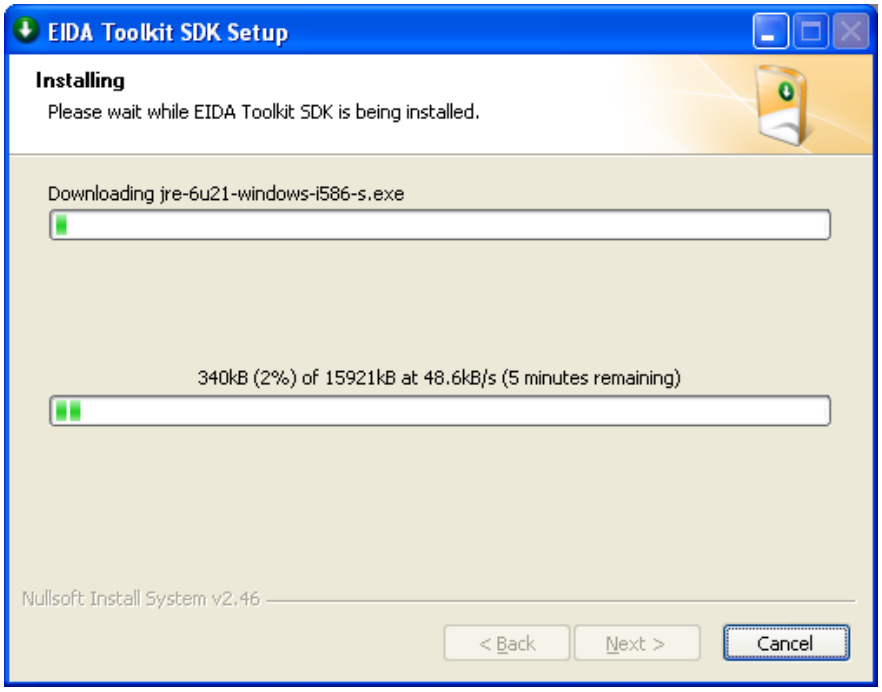


Figure 7: Downloading Java Runtime

- After downloading the Java Runtime, the setup wizard will automatically launch the java runtime installation wizard.

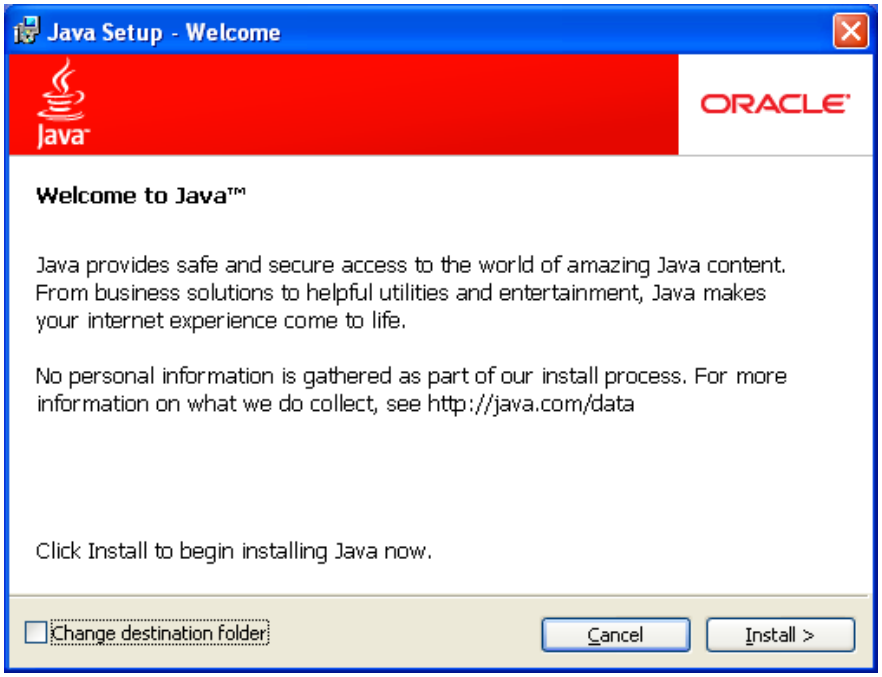
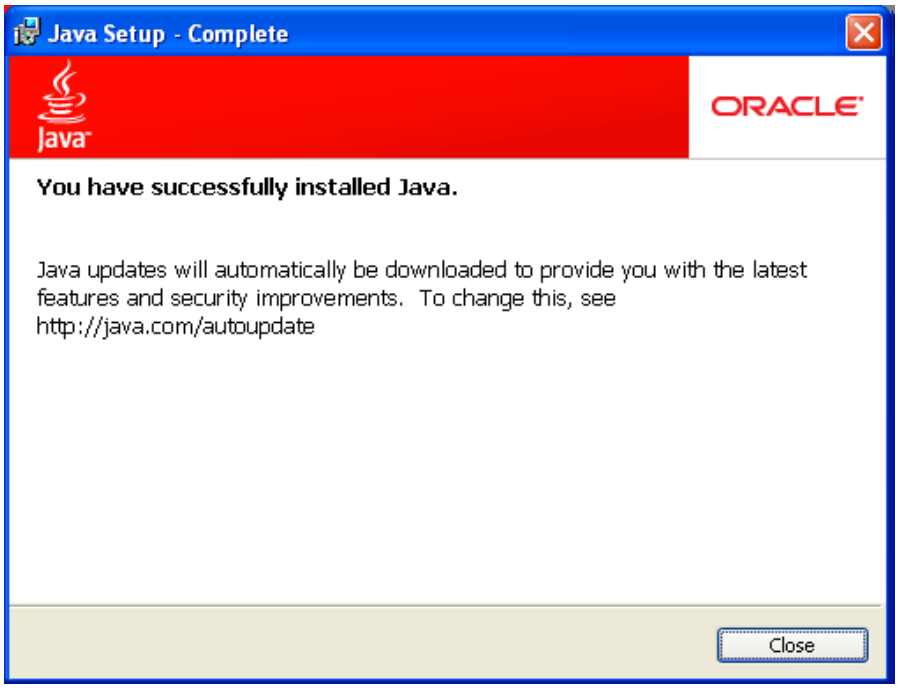


Figure 8: Java Runtime Installation

- Follow the Java Runtime installation wizard until it's completed successfully and then click Close to get back to EIDA toolkit SDK setup wizard (
-
-



- **Figure 9**).

Figure 9: Java Runtime Installation – Complete

- Setup wizard then checks if the .Net framework v3.5 is installed, if it's not installed the wizard will start downloading it (
-
-

- **Figure 10**).

- If the .Net framework is already installed, the setup wizard will skip this step, or cancel the download and continue installation to install it later.

- After downloading the .Net framework the setup wizard will silently install it, and continue the toolkit installation.

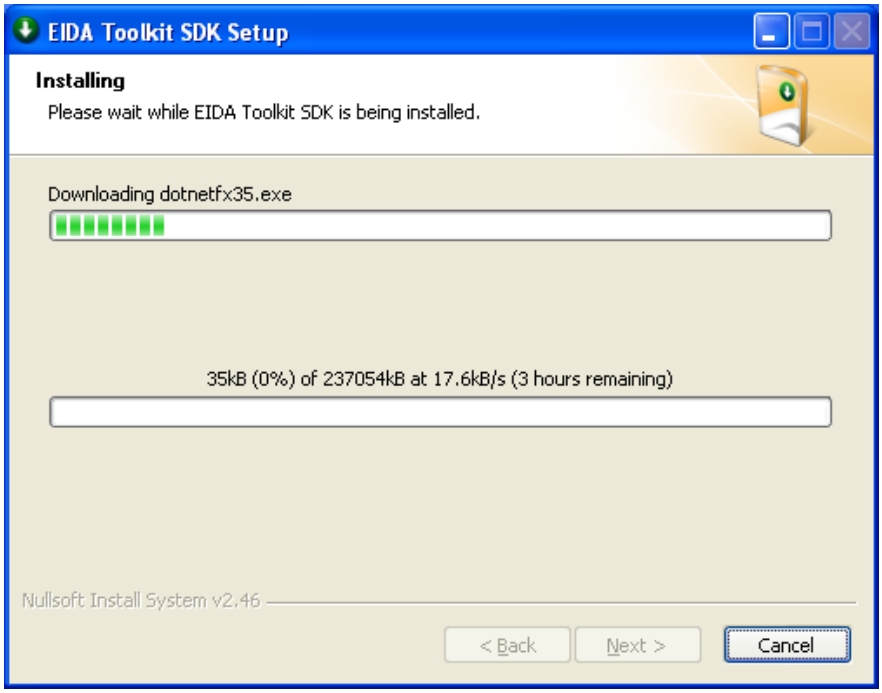


Figure 10: Downloading .Net Framework 3.5

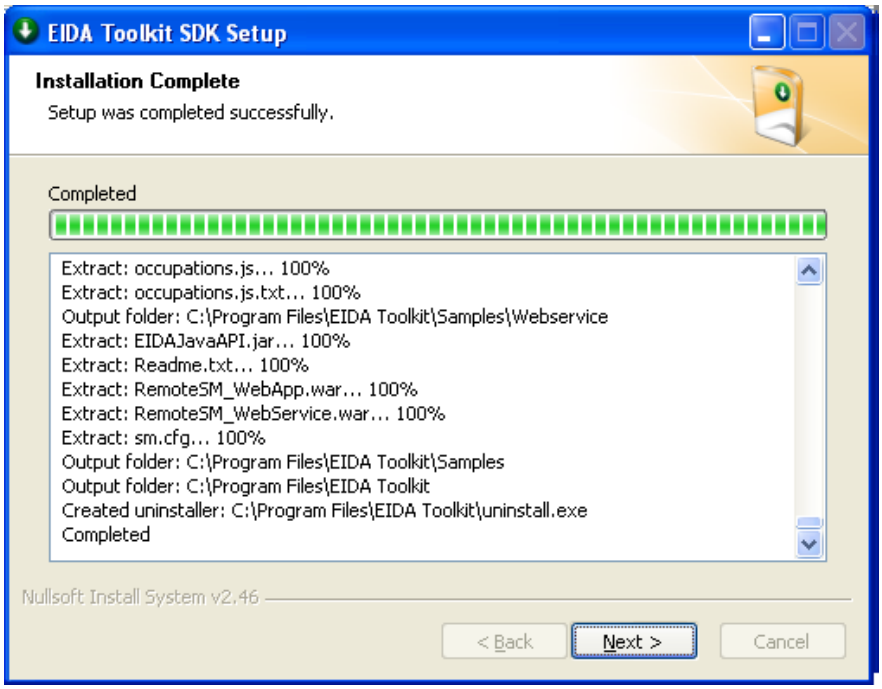


Figure 11: Installation Details

- The setup wizard will install the selected toolkit components and register components if needed displaying the installation details (
-
-

- **Figure 11**).

- Click “Next” to finish, the setup wizard will display its Finish screen (
-
-

- **Figure 12**).

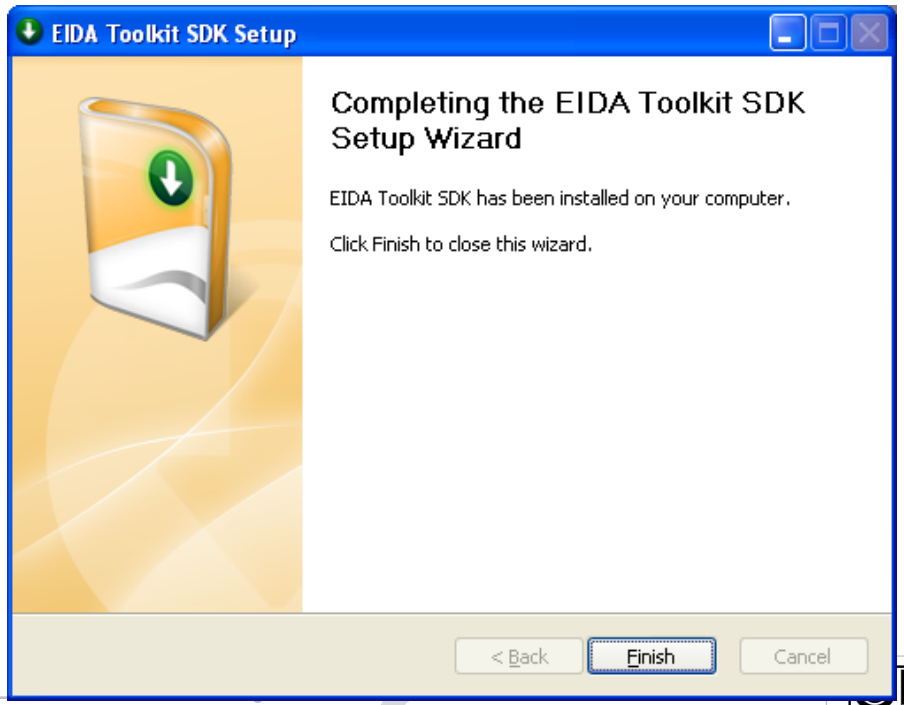


Figure 12: Installation Complete

4.3 Post-Installation Test

To test if the installation has been done correctly, below tests can be run only if Samples component option was selected while installation.

- Go to the installation directory (e.g., C:\Program Files\EIDA Toolkit)
- Navigate to “Samples\Desktop” directory
- Open either “dotNet” or “Java” directory
- Run the file “EIDADesktopSampleApplication.exe” or “launch.bat”
- EIDA Toolkit Sample Application will start asking to select a card reader
- If card reader select dialog box appears, then the installation has been done successfully

4.4 Uninstalling EIDA Toolkit SDK

EIDA toolkit SDK can be uninstalled from “Add/Remove programs” in control panel

Find the EIDA toolkit SDK then right click and select uninstall from the drip down menu, another wizard will start guiding you through the removal operation.

5 Configuration of Toolkit applications

This section describes various configurations required to run the Test applications.

5.1 Configuration of Local Secure Messaging Modules

Desktop applications use Secure Messaging modules (sm) in local mode. All sm modules will be described in the sm.cfg file as described below.

The file sm.cfg contains the list of configured secure messaging modules' parameters.

sm.cfg

```
[SM_Modules]
####### SAGEM_SAM 1, SAFENET_LUNA_HSM 2, LOGICA_SOFTWARE_HSM=3
ID_SM_Name=3
PKI_SM_Name=3
MOC_SM_Name=3

5=Another_SM.dll

[Data Signing Certificates]
PATH=C:\Program Files\EIDA Toolkit\Libs\SigningCerts

[SAM]
PIN=0123
ATR1=3B 7F 11 00 00 80 41 00 57 4A 2D 49 44 4D 36 34 83 7F 90 00
ATR2=3B FF 95 00 FF 40 0A 80 31 E8 73 F6 21 13 67 4A 47 48 60 31 42 00
ATR3=3B 78 18 00 00 01 53 41 4D 20 45 41 55

[HSM]
PIN=1234567

[UAE Card]
NUMBER=5
ATR1=3B 6A 00 00 80 65 A2 01 30 01 3D 72 D6 41
ATR2=3B 6A 00 00 80 65 A2 01 31 01 3D 72 D6 41
ATR3=3B 7A 95 00 00 80 65 A2 01 30 01 3D 72 D6 41
ATR4=3B 7A 95 00 00 80 65 A2 01 31 01 3D 72 D6 41
ATR5=3B 8A 80 01 80 65 A2 01 31 01 3D 72 D6 41 A5

[Test Card]
NUMBER=5
ATR1=3B 6A 00 00 80 65 A2 01 30 01 3D 72 D6 41
ATR2=3B 6A 00 00 80 65 A2 01 31 01 3D 72 D6 41
ATR3=3B 7A 95 00 00 80 65 A2 01 30 01 3D 72 D6 41
ATR4=3B 7A 95 00 00 80 65 A2 01 31 01 3D 72 D6 41
ATR5=3B 8A 80 01 80 65 A2 01 31 01 3D 72 D6 41 A5
```

[SM_Modules]: this section contains the secure messaging modules.

- The ID_SM_Name attribute refers to the secure messaging module used with the ID Applet.
- The PKI_SM_Name attribute refers to the secure messaging module used with the PKI Applet.

- The MOC_SM_Name attribute refers to the secure messaging module used with the MOC Applet.

The value of these three attributes can be:

- 1: this means that the SAGEM_SAM is used.
- 2: the SAFENET_LUNA_HSM is used.
- 3: the LOGICA_SOFTWARE_HSM is used (to be used with test cards only).
- 4,5,6,7 …: another secure messaging module is used. This extra sm module is implemented on a separate DLL.

For example, if the SM module used with the PKI applet is implemented on the Another_SM.dll. the PKI_SM_Name will be set to 4. Then another attribute is added under SM_Modules to reference the new sm module which means “4=Another_SM.dll”.

Note that the Another_SM.dll library must be located on the same directory as the UAE_IDCardLib.dll.

NOTE: the section “Another_SM” in the DLL name “Another_SM.dll” could be any name for the DLL

[Data Signing Certificates] this section contains only the “PATH” attribute which holds the path for the directory containing EIDA data signing certificates that are used by the Toolkit to validate signatures in the public data files.

[SAM] this section contains the following attributes:

- PIN attribute: It is used only when a one of the three attributes XXX_SM_Name points to the SAGEM_SAM. The PIN is provided by EIDA.
- ATR attribute: this attribute is used to recognize the EIDA SAM cards, it’s possible to provide 3 different values; ATR1, ATR2, and ATR3

[UAE Card]: this section identifies EIDA live ID cards ATRs

- Number attribute: used as a counter for the number of the configured ATRs for the toolkit to read from the current section.
- ATR attribute: each attribute contains an ATR for live ID cards

[Test Card]: this section identifies EIDA test ID cards ATRs

- Number attribute: used as a counter for the number of the configured ATRs for the toolkit to read from the current section.
- ATR attribute: each attribute contains an ATR for test ID cards

The sm.cfg file must be located on the same working directory of the desktop application or under the config folder of the working directory; otherwise the toolkit will use the default sm.cfg file that’s located in Libs folder of the toolkit installation directory.

5.2 Deployment and Configuration of Secure Messaging Web Service

The Secure Messaging Web service is a server side component works over HTTP(s) protocol and offers remote access to EIDA secure messaging modules (SAM or HSM) in order for EIDA Toolkit client components such as Toolkit web components (for web applications)or Toolkit APIs (for desktop applications) to consume the services. This web service is required in case of using some of the card functions that requires either secure messaging or cryptography authentication in order to read sensitive data from the card.

The web service is based on the toolkit APIs, therefore EIDA toolkit must be installed on the server hosting the web service.

The deployment differs according the implementation type of the web service; the toolkit offers two implementations; Java and dotNet, the below subsections describes the deployment of each respectively.

5.2.1 Java based web service

A java servlet component to be deployed on a servlet container supporting java 6 (the most common option is Apache Tomcat ver. 6 or higher). Supported platforms are Windows 2003 / Win 2008.

Follow the below steps required to deploy the web service on Tomcat:

1. Copy the WAR file RemoteSecureMessagingService.war from the path “toolkit installation folder”\WebServices\Java to the Tomcat folder “webapps”
2. Copy the file “Logger.ini” from the path “toolkit installation folder”\WebServices” to the Tomcat folder “bin”, then configure the log file path and the log level in this file
3. Start Tomcat which will extract the .war file.
4. Edit the file “webapps\RemoteSecureMessagingService\WEB-INF\web.xml” as below
    - Change the value of the parameter `SECURE_MESSAGING_MODULES_TYPE` to the required SM module you want to use. This value can be SOFT, SAM, or HSM.
    - Configure the password\PIN of the SM in the parameter `SECURE_MESSAGING_MODULES_PASSWORD` (this parameter is ignored in case of SOFT)

5.2.2 .Net Web Service

An ASP.NET HTTP responder can be deployed on Microsoft IIS 6 or higher as a pre-requisite.

Follow the below steps to deploy the web service on IIS 7:

1. Copy the folder RemoteSecureMessagingService from the path "toolkit installation folder"\WebServices\dotNet to the folder "C:\inetpub\wwwroot" which is the web site physical path.
2. Open the IIS Manager, and expand the nodes on the left hand side till reaching the node representing the desired web site.
3. The folder "RemoteSecureMessagingService" should be displayed as a need under the web site node , right clock on it then select "Convert to Application"
4. Copy the file "Logger.ini" from the path "toolkit installation folder"\WebServices" to the folder "windows installation folder"\System32\inetsrv", and then configure the log file path and the log level.
5. Edit the file "RemoteSecureMessagingService\Web.config" as explained in step 4 of the Java web service section.
6. Configure the IIS worker process to write on the log file (please refer to the IIS documentation to do so).

5.2.3 General notes

- The web service automatically detects all the connected SAMs and performs the load balance between them automatically.
- In case of using multiple SAM cards then all of them must have the same PIN
- For load balancing installations, Session replication has to be configured between server nodes (please refer to your application server manual for session replication configuration).

5.3 Deployment and Configuration of Sample Web application

This section provides the steps required to deploy the sample web application used to demonstrate the usage of the Toolkit zero footprint web components. The toolkit provides two types of sample implementations namely Java and .Net.

5.3.1 Java based web application

The sample java web application is built with basic JSP and servlets supporting java 6 (Apache Tomcat 6 or higher) in the supported Windows 2003 / Win 2008 environments.

Follow the below steps to deploy the web service on Tomcat:

1. Copy the WAR file ZFDemoSite.war from the path of toolkit installation folder ”Samples\Website\Java” to the Tomcat folder “webapps”.
2. Start Tomcat which will extract the .war file.
3. Edit the file “webapps\ ZFDemoSite\WEB-INF\web.xml” as below
    - Change the value of the parameter `ValidationMethod` to the preferable method. Value can be OCSP, CRL, or CDP. Note that CDP is not supported by EIDA certificates currently.
    - In case the validation method configured is OCSP, OCSP Url should be configured. Change the value of the `OCSP_URL` parameter to the OCSP service URL to be used to validate EIDA certificates.
    - CA certificates are also required in case of OCSP validation. By default, EIDA Live CA certificates are placed need in the “webapps\ZFDemoSite\ca_certs” folder.
    - If CRL is the validation method configured, then make sure that the “webapps\ZFDemoSite\crls” folder has the latest CRL files to be requested from EIDA.
4. Public data file signatures validation is configured to validate against certificates are placed in the folder “webapps\ZFDemoSite\data_signing_certs”. By default, this folder contains the certificates required by EIDA Live ID card. More certificates can be added if available from EIDA.

5.3.2 .Net web application

The sample ASP.NET web application can be deployed on Microsoft IIS 6 or higher.

Follow the below steps to deploy the web application on IIS:

1. Copy the folder “ZFDemoSite” the path of toolkit installation folder. ”Samples\Website\dotNet to the folder “C:\inetpub\wwwroot” which is the web site physical path.
2. Open the IIS Manager, and expand the nodes on the left hand side till the node representing the desired web site.

3. The folder “ZFDemoSite” should be displayed under the web site node. Right click on it then select “Convert to Application”.

4. Edit the file “ZFDemoSite\Web.config” as explained in step 3 of the Java web site section above.

Configure the IIS worker process to write on the log file (please refer to the IIS documentation to do so).