



**Atmel AT88CKECCSIGNER Provisioning Signer Module Kit**

## Introduction

The Atmel® Signer Module Utility application provides an easy and secure method to create an intermediate certificate authority for provisioning the Atmel® ECC-based CryptoAuthentication™ devices. The Intermediate Certificate Authority is created by having the Root Certificate Authority Sign the Public Key of the Signer Module. This can then be used to sign end devices at the subcontractor or to sign firmware before download or deployment. This document describes the usage of the Atmel Signer Module Utility application.

## Features

- Un-configured Signer Module Flow to create your own Intermediate Certificate Authority (CA)
- Configured Signer Module Flow to create additional intermediate Certificate Authorities (CA)

## Table of Contents

---

<b>Un-configured Signer Module Flow .....</b>	<b>3</b>
Step 1      Start the Signer Module Utility application .....	3
Step 2      Insert an Un-configured Signer Module .....	3
Step 3      Signer Module Configuration.....	4
Step 4      Signer Module Advanced Configuration.....	6
Step 5      Signer Module Load Backup File .....	7
Step 6      Configure Signer Module .....	8
Step 7      Save The Signer Module Source Code Files .....	10
Step 8      Saving the Signer Module file as a Backup.....	11
Step 9      Signer Module Additional Information .....	12
<b>Configured Signer Module Flow .....</b>	<b>13</b>
Step 1      Start the Signer Module Utility Application .....	13
Step 2      Insert Configured Root Module .....	14
Step 3      Signer Module Additional Information .....	14
<b>Atmel Evaluation Board/Kit Important Notice and Disclaimer .....</b>	<b>17</b>
<b>Revision History.....</b>	<b>17</b>

## Un-configured Signer Module Flow

### Step 1 Start the Signer Module Utility application

Start the Signer Module Utility application by selecting the Signer Module Utility application from the following Microsoft Window Start Menu location:

- ▶ Select the **Start Menu > All Programs > Atmel Secure Products > Provisioning Kits > and then Signer Module Utility.**

The **Atmel Signer Module Utility** application window displays as shown below:

Figure 1. Signer Module Utility Application Main Window



### Step 2 Insert an Un-configured Signer Module

1. Insert an un-configured Signer Module from the AT88CKECCSIGNER Provisioning Signer Module Kit. The Signer Module is read by the Signer Module Utility application, and information about the Signer Module is displayed on the Signer Module Utility application main window.
2. Insert a configured Root Module from the AT88CKECCROOT kit.



The Root Module must have a unique key configuration completed.



Keep the Root and Signer Modules inserted in the computer until the Signer Module configuration has been completed.



The Signer Module is configured using the already configured inserted Root Module.

3. Select **Next >** to continue to the Signer Module configuration.


**Figure 2. Sample Un-configured Signer Module Main Window**



### Step 3 Signer Module Configuration

1. Provide the Signer Module configuration information as shown in the figure below.



Move the mouse cursor over the  image to display help information about the Signer Module configuration information.

#### – Signer Configuration

- **Module Description (Not Required)**
  - The description the Signer Module to be configured. Is referenced in the future.
  - Maximum length is 63 alpha-numeric characters.
- **Certificate Common Name (Required)**
  - The Signer Module certificate common name.
  - The Signer Module certificate common name is required for the Signer Module's X.509 certificate.
  - The Signer Module certificate common name is used to uniquely identify the Signer Module's X.509 certificate.
  - Maximum length is 60 alpha-numeric characters.

- **Password (Optional): (Not Required)**
    - Used to access and use the Signer Module in the provisioning production flow.
    - Maximum length is 31 alpha-numeric characters.
  - **Device Configuration**
    - **Certificate Common Name (Required)**
      - The device certificate common name.
      - The device certificate common name is required for the device's X.509 certificate.
      - The device certificate common name is used to uniquely identify the device's X.509 certificate.
      - Maximum length is 63 alpha-numeric characters
2. Select **Advanced...** to open the **Signer Module Utility - Advanced Configuration Information** dialog box.
  3. After entering the Signer Module configuration information, select **Commit >** to configure the Signer Module. See Step 6 for more information.

**Figure 3. Signer Module Configuration**

The screenshot shows the 'Atmel Signer Module Utility' window, Version 1.0.0. The main text reads: 'Please provide the following Signer Module configuration information.' The window is divided into two main sections: 'Signer Configuration' and 'Device Configuration'. Each section has a blue question mark icon in the top right corner. The 'Signer Configuration' section includes:
 

- Module Description (max 63 characters):** Example ATECC508A Signer
- Certificate Common Name (max 60 characters):** Example ATECC508A Signer
- Password (Optional):** (empty field)
- Confirm Password (Optional):** (empty field)

 The 'Device Configuration' section includes:
 

- Certificate Common Name (max 63 characters):** Example ATECC508A Device

 At the bottom right of the configuration area is an 'Advanced...' button. Below the configuration area, it says 'Press the Commit button to configure the Signer Module.' At the very bottom, there is a footer with the URL 'www.atmel.com/eccroot-sign', and three buttons: '< Back', 'Commit >', and 'Exit'. Orange arrows point to the question mark icons and the 'Advanced...' button.

---


## Step 4 Signer Module Advanced Configuration

---

Provide the following Signer Module advanced configuration information.

1. Click on **Load Signer Module Backup File...** to load the Signer Module backup file. See Step 5 for more information.



Move the mouse cursor over the  image to display help information about the Signer Module configuration information.

- **Signer Module Configuration**

- **Signing Limit (1 – 2097151) (Required)**

- The number of times this Signer Module can sign an ECC device during production.
    - The signing limit can be used to limit the number of devices that are signed by the Signer Module.
    - Maximum limit is 2097151

- **Certificate Expiration (Required)**

- Expiration date in years since the issue date.
    - The expiration date is added to the Signer Module's X.509 certificate.
    - Determines how many years this Signer Module can sign device X.509 certificates in the provisioning production flow.
    - Possible values:
      - None: No expiration date specified.
      - The number of years before the Signer Module's X.509 certificate expires.

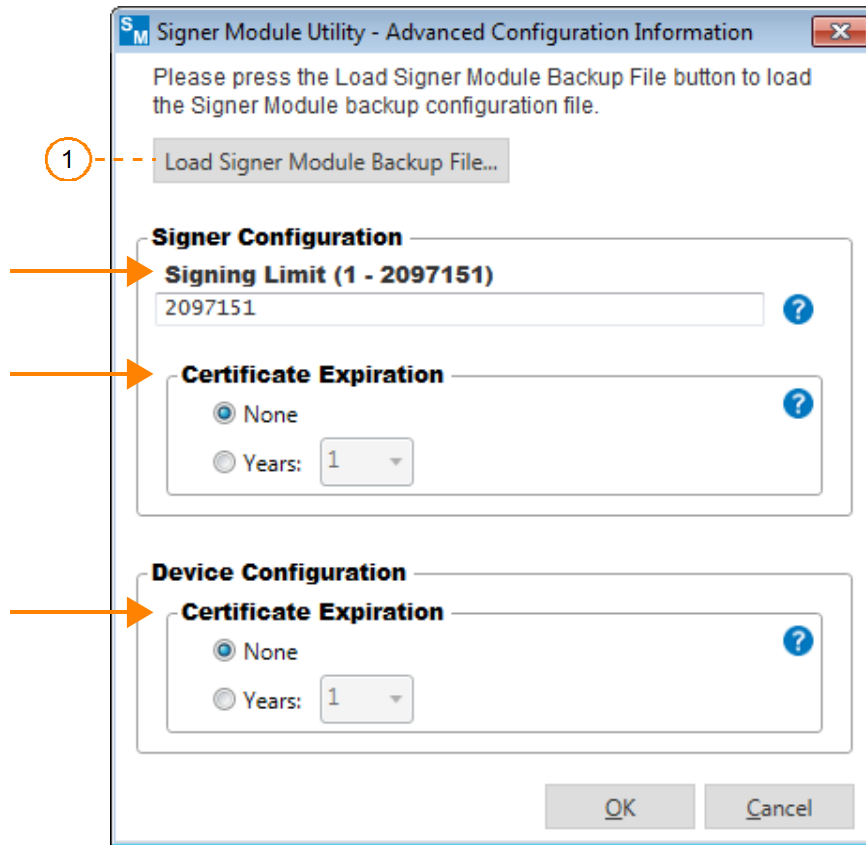
- **Device Module Configuration**

- **Certificate Expiration (Required)**

- Expiration date in years since the issue date.
    - The expiration date is added to the device's X.509 certificate.
    - Possible values:
      - None: No expiration date specified.
      - Years (1 – 31): The number of years before the device's X.509 certificate expires.

2. Click on **OK** to save any changes to the Signer Module advanced configuration.

Figure 4. Signer Module Advanced Configuration



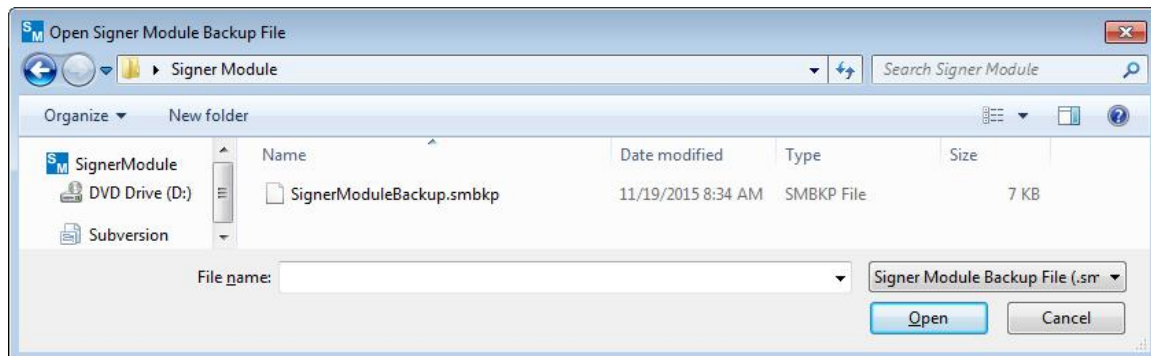
## Step 5 Signer Module Load Backup File

The Signer Module Load Backup File capability is used to create a Signer Module with the same certificate configuration as a previous configured Signer Module.



This does *not* duplicate the private key, but only the certificate definitions. The original Signer Module configuration process saved the backup file.

Figure 5. Example Open Signer Module Load Backup File Dialog Box



After the Signer Module load backup file has been loaded, the Signer Module configuration information is loaded from the backup file, and some editable fields are disabled.

Figure 6. Example Signer Module Configuration Information After The Backup File Is Loaded

The screenshot shows the 'Atmel Signer Module Utility' window, Version 1.0.0. The window title is 'Atmel® Signer Module Utility'. The main content area has a header with the Atmel logo and the title 'Signer Module Utility'. Below the header, it says 'Please provide the following Signer Module configuration information.' There are two main sections: 'Signer Configuration' and 'Device Configuration'. The 'Signer Configuration' section includes fields for 'Module Description (max 63 characters):' (Example: ATECC508A Signer), 'Certificate Common Name (max 60 characters):' (Example: ATECC508A Signer 0002), 'Password (Optional):', and 'Confirm Password (Optional):'. The 'Device Configuration' section includes a field for 'Certificate Common Name (max 63 characters):' (Example: ATECC508A Device). There is an 'Advanced...' button to the right of the 'Device Configuration' section. At the bottom, there is a 'Commit' button and an 'Exit' button. A footer contains the URL 'www.atmel.com/eccroot-sign' and navigation buttons '< Back', 'Commit >', and 'Exit'.

## Step 6 Configure Signer Module

1. After entering the Signer Module configuration information, click on **Commit >** to configure to the Signer Module. The Signer Module configuration process starts.



It is very important that the configured Signer Module be stored in a save place. The Signer Module is part of the trusted certificate authority within the Atmel Secure Provisioning System.

2. Follow the directions to configure the Signer Module.



Figure 7. Signer Module Configuration

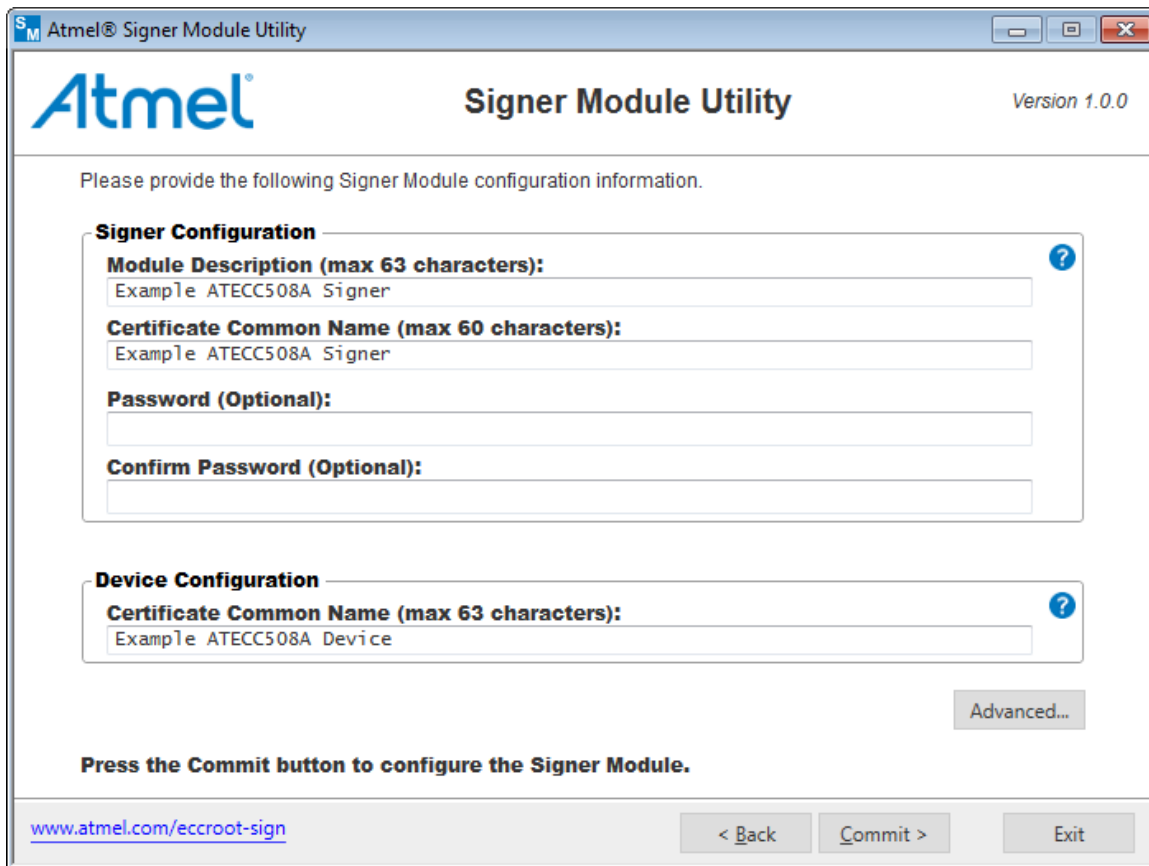


Figure 8. Signer Module Configure Warning Dialog box

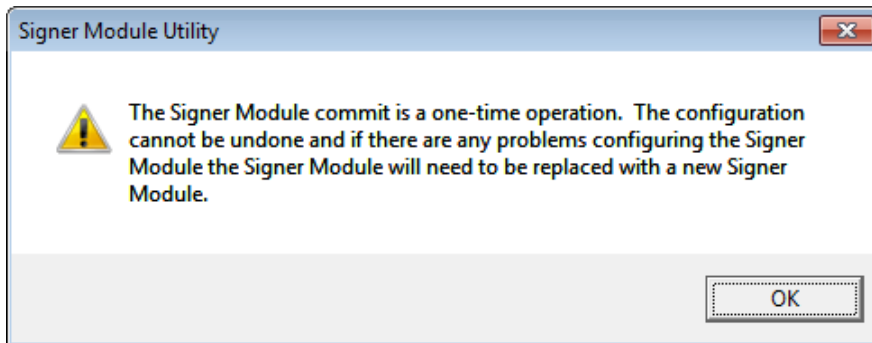
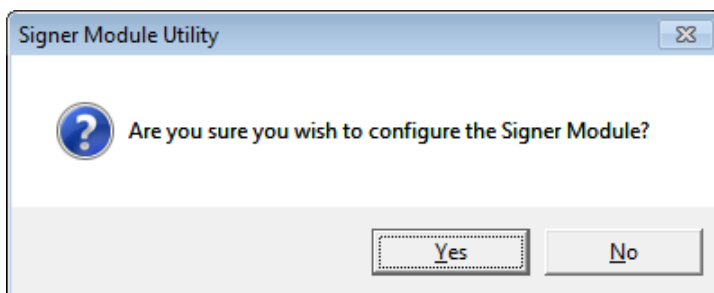
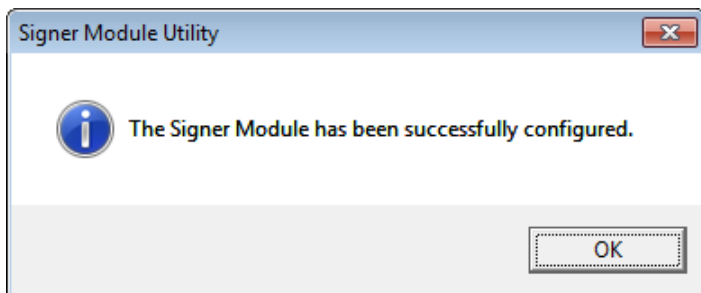


Figure 9. Signer Module Configure Question Dialog Box



**Figure 10. Signer Module Successful Configuration Dialog Box**



3. After the Signer Module is successfully configured, it asks to save the Signer Module source code files used during the creation of the customized provisioning firmware to work with the Provisioning Production Server. See Step 7 for more information.
4. After a Signer Module backup file is saved, options to save additional configured Signer Module information are provided. See Step 9 for more information.

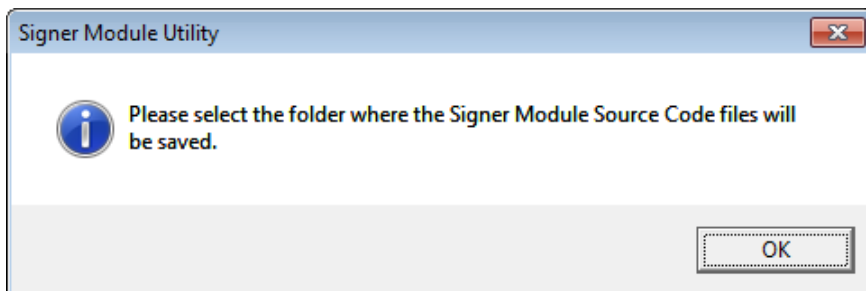
---

### **Step 7 Save The Signer Module Source Code Files**

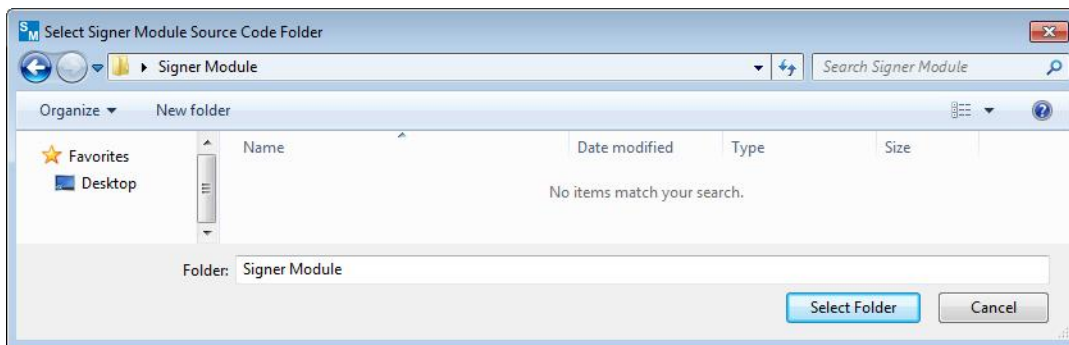
---

After the Signer Module has been configured, the option to save the Signer Module source code files is given. Start the save the Signer Module source code files process.

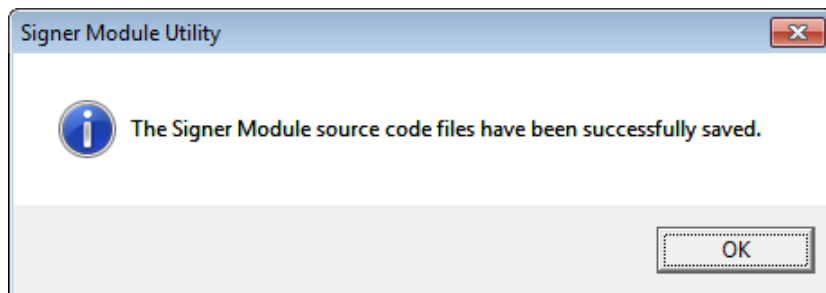
**Figure 11. Save the Signer Module Source Code Files Dialog Box**



**Figure 12. Select Signer Module Source Code Folder Dialog Box**



**Figure 13. Signer Module Source Code Files Saved Successfully Dialog Box**



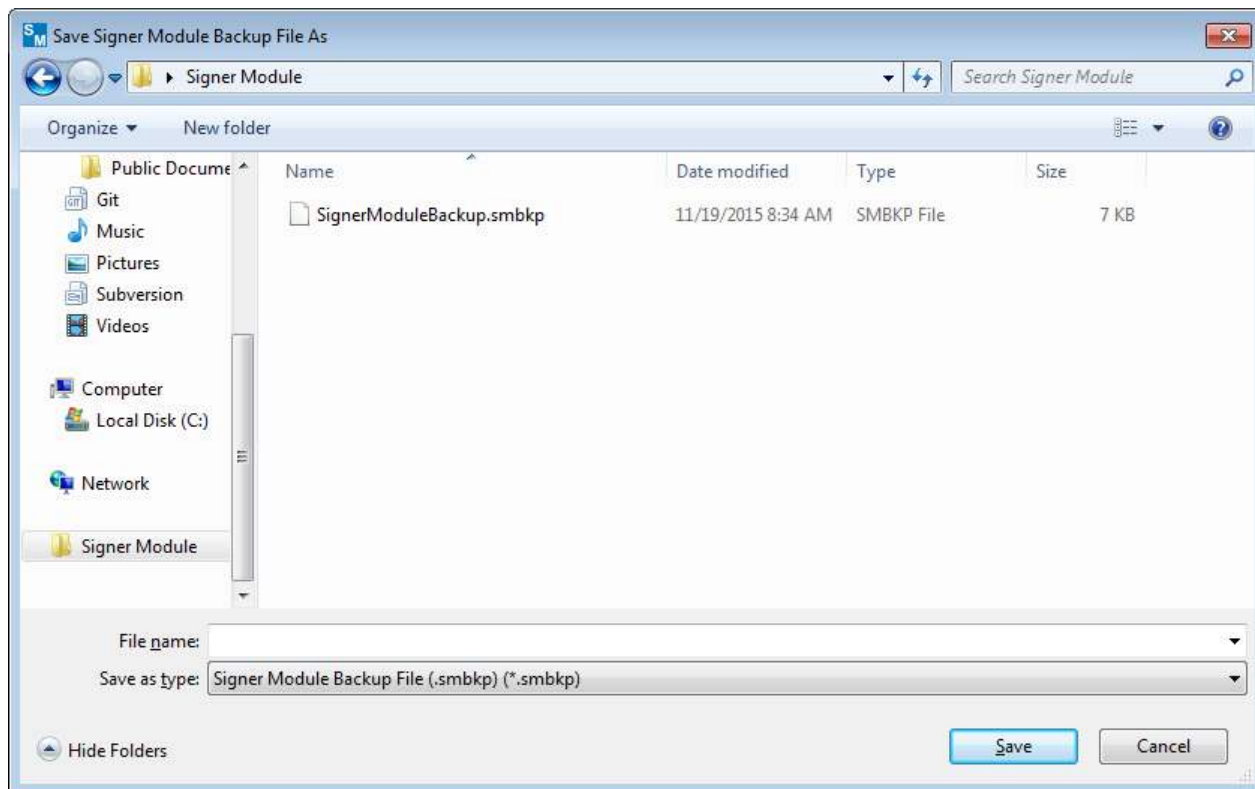
### Step 8 Saving the Signer Module file as a Backup

The Signer Module Save Backup File capability is used to save a Signer Module's configuration. The backup is used to create additional Signer Modules with the same certificate definitions at a later date and time.



It is extremely important to save the Signer Module configuration backup file.

**Figure 14. Example Save Signer Module Backup File Dialog**



## Step 9 Signer Module Additional Information

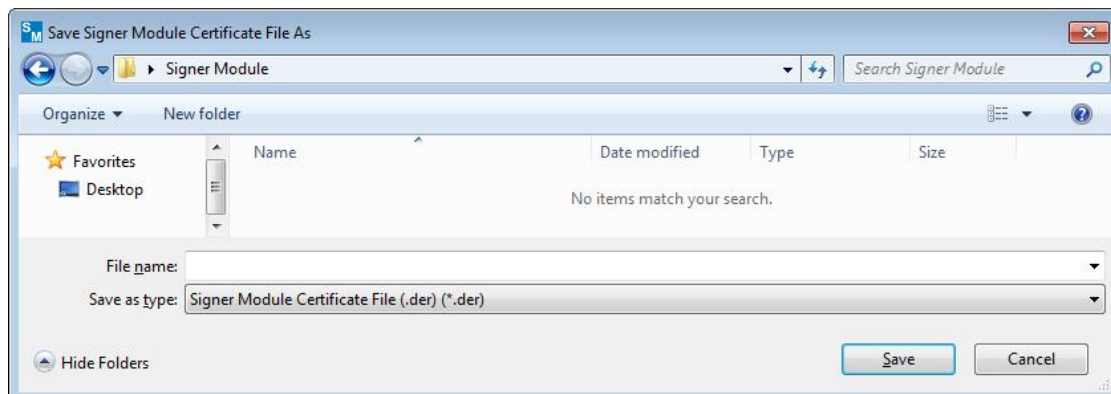
The configured Signer Module Additional Information allows the Signer Module operations to be performed.

- **Save Signer Module Backup File...** button:
  - Saves the Signer Module backup file. See Step 8 for more information.
- **Save Signer Module Certificate File...** button:
  - Saves the Signer Module X.509 certificate file. This certificate file is used to verify that an AT88CKECCSIGNER Signer Module was created with the inserted configured Root Module.
- **Save Source Code Files...** button:
  - Starts the process to save the Signer Module source code files. See Step 7 for more information.

Figure 15. Configured Signer Module Additional Information Dialog Box



Figure 16. Example Save Signer Module Certificate File Dialog Box



Note: This certificate file can verify that a Signer Module was created with the inserted configured Root Module.

## Configured Signer Module Flow

After the original Signer Module has been configured, the option to create as many Signer Modules that are needed is given. Please follow the directions carefully.

### Step 1 Start the Signer Module Utility Application

Start the Signer Module Utility application by selecting the Signer Module Utility application from the Microsoft Window Start Menu location:

- ▶ Select the **Start Menu > All Programs > Atmel Secure Products > Provisioning Kits >** and then **Signer Module Utility**.

The **Atmel Signer Module Utility** window displays:

Figure 17. Signer Module Utility Application Main Window



## Step 2 Insert Configured Root Module

1. Insert a configured Signer Module from the AT88CKECCSIGNER Kit.
2. Insert the Root Module (from the AT88CKECCROOT kit) that was used to configure the Signer Module in the first place.

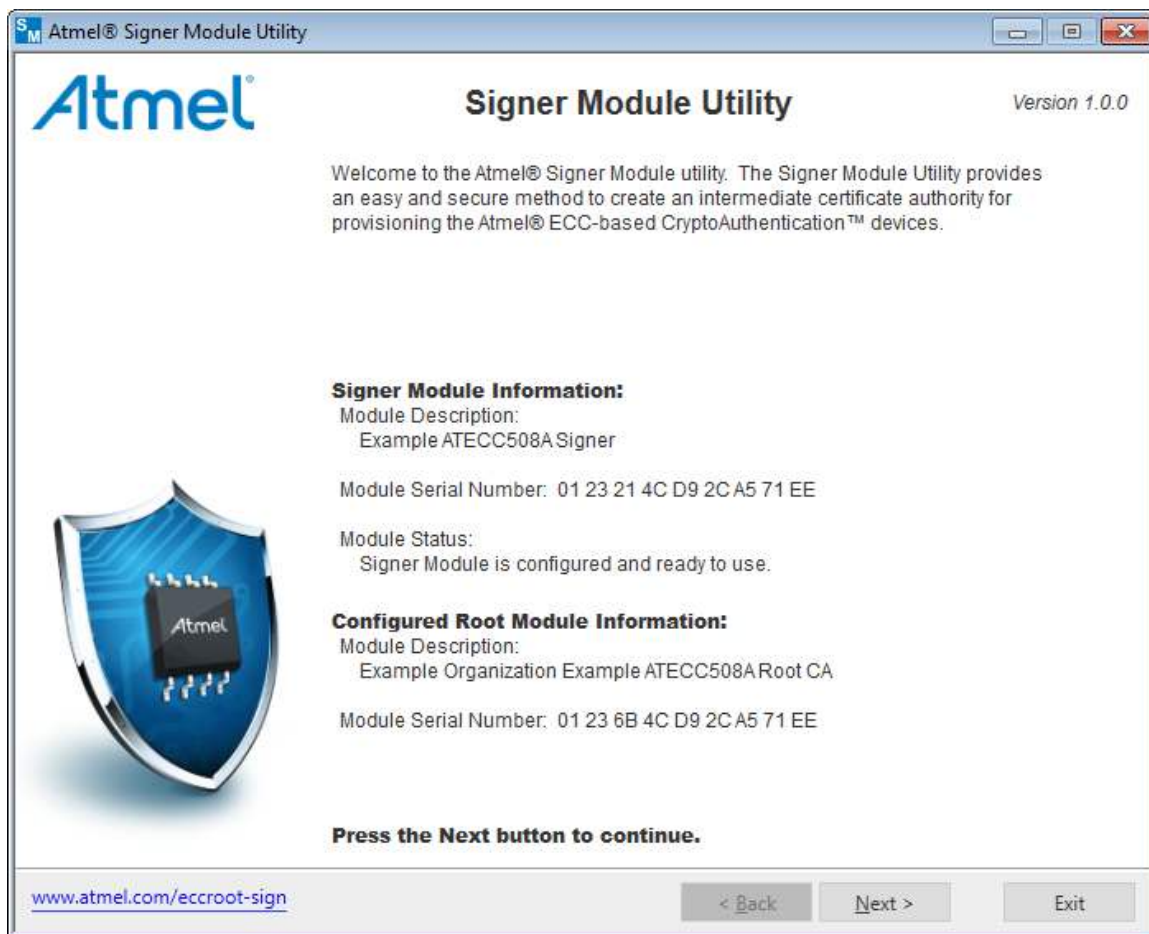
The Signer Module is read by the Signer Module Utility application, and information about the Signer Module is displayed on the Signer Module Utility application main window.



Keep the Root and Signer Modules inserted in the computer until the Signer Module configuration has been completed.

3. Click on **Next >** to continue to the Signer Module configuration

Figure 18. Sample Configured Signer Module Main Window



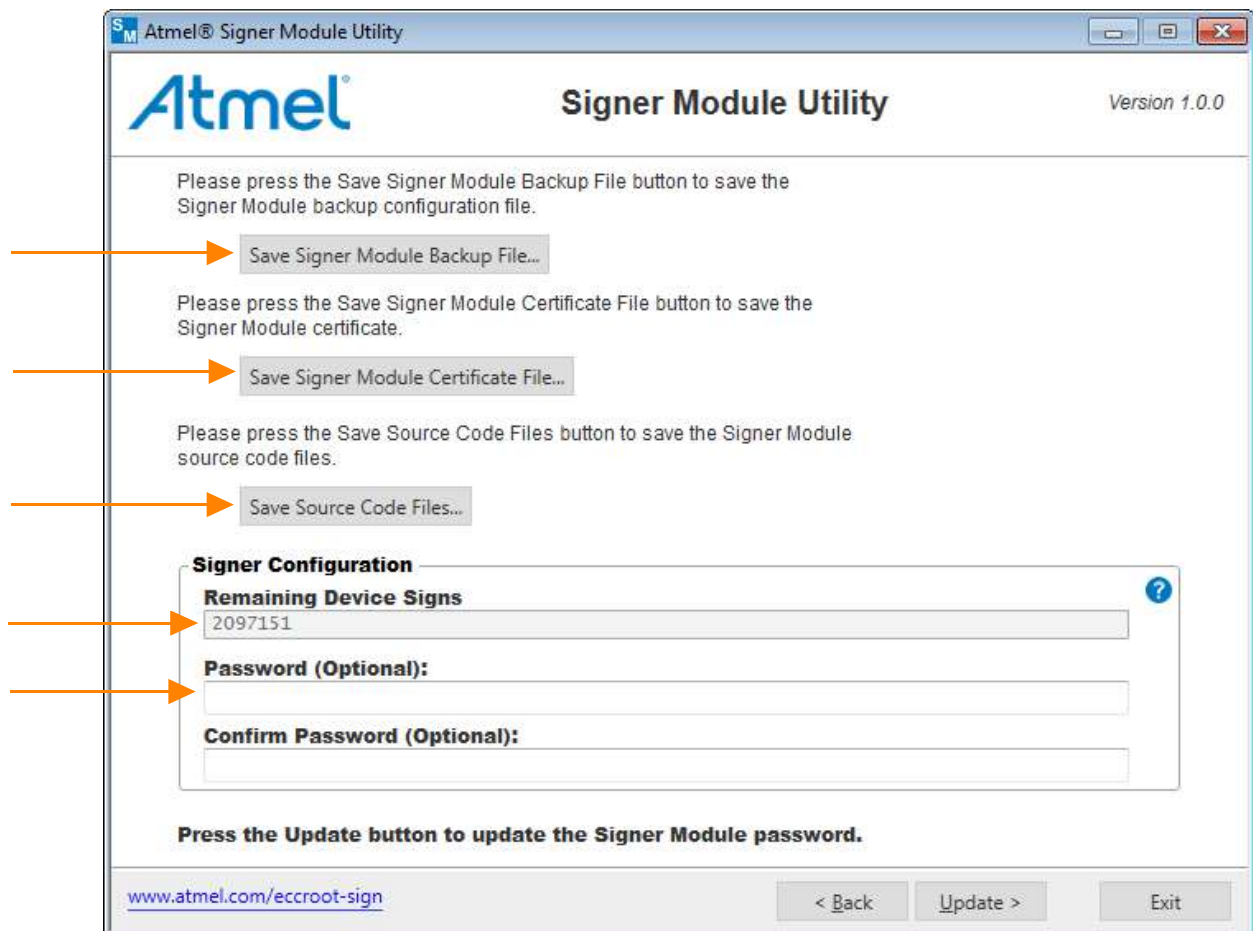
## Step 3 Signer Module Additional Information

The configured Signer Module Additional Information allows the Signer Module to perform its operations. Select **Update >** to update the Signer Module password.

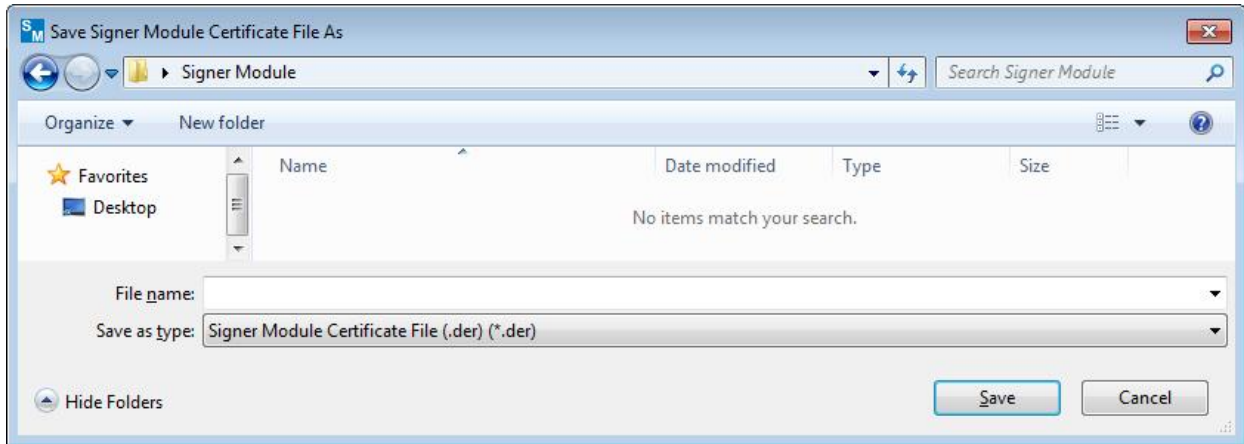
- **Save Signer Module Backup File...** button:
  - Saves the Root Module backup file. See Step 8, “Saving the Signer Module file as a Backup” from the previous section for more information.

- **Save Signer Module Certificate File...** button:
  - Saves the Signer Module X.509 certificate file. This certificate file is used to verify that an AT88CKECCSIGNER Signer Module was created with the inserted configured Root Module.
- **Save Source Code Files...** button:
  - Starts the process to save the Signer Module source code files. See Step 7, “Save The Signer Module Source Code Files” from the previous section for more information.
- **Signer Configuration:**
  - **Remaining Device Signs:**
    - The remaining number of times this Signer Module can sign an ECC device during production.
  - **Password (Optional):**
    - Allows the Signer Module password used to access and use the Signer Module in the provisioning production flow to be changed.
    - Maximum length is 31 alpha-numeric characters.
    - To remove the password in the provisioning production flow, set the password to an empty password and select **Update >** to update the Signer Module password.

Figure 19. Configured Signer Module Additional Information Dialog Box

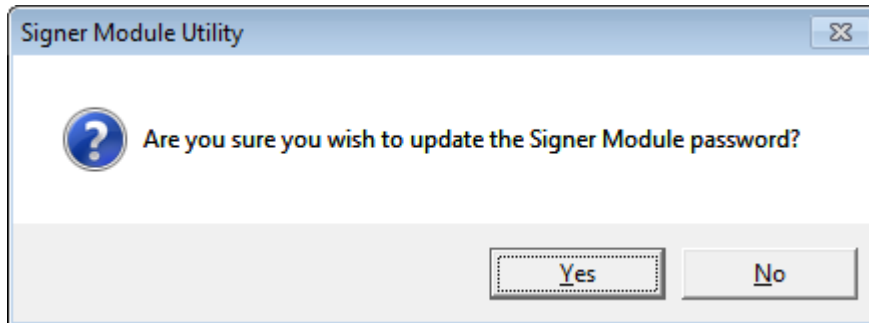


**Figure 20. Example Save Signer Module Certificate File Dialog Box**

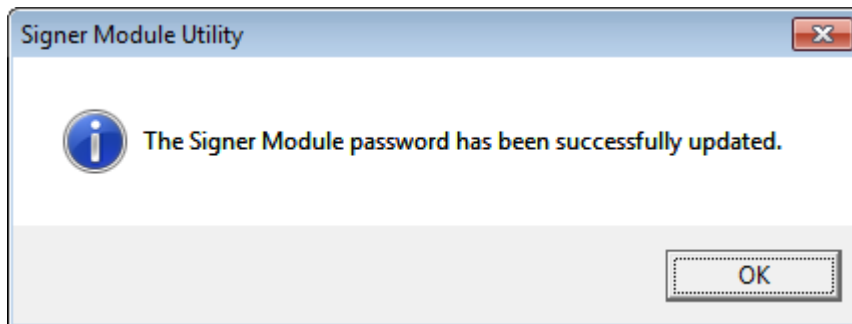


Note: This certificate file can verify that a Signer Module was created with the inserted configured Root Module.

**Figure 21. Update Signer Module Password Question Dialog Box**



**Figure 22. Signer Module Password Updated Successfully Dialog Box**





## Atmel Evaluation Board/Kit Important Notice and Disclaimer

This evaluation board/kit is intended for user's internal development and evaluation purposes only. It is not a finished product and may not comply with technical or legal requirements that are applicable to finished products, including, without limitation, directives or regulations relating to electromagnetic compatibility, recycling (WEEE), FCC, CE or UL. Atmel is providing this evaluation board/kit "AS IS" without any warranties or indemnities. The user assumes all responsibility and liability for handling and use of the evaluation board/kit including, without limitation, the responsibility to take any and all appropriate precautions with regard to electrostatic discharge and other technical issues. User indemnifies Atmel from any claim arising from user's handling or use of this evaluation board/kit. Except for the limited purpose of internal development and evaluation as specified above, no license, express or implied, by estoppel or otherwise, to any Atmel intellectual property right is granted hereunder. ATMEL SHALL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RELATING TO USE OF THIS EVALUATION BOARD/KIT.

ATMEL CORPORATION  
1600 Technology Drive  
San Jose, CA 95110  
USA

## Revision History

Doc Rev.	Date	Comments
8969A	12/2015	Initial document release.

